



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACION Y
CONTROL**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL DE POSICIÓN PARA EL CUADRICÓPTERO
AR.DRONE 2.0 DE PARROT”**

COLOMA VALLEJO GERMMY FRANKLIN

OYOS PIARPUESAN KID COLEMAN

DIRECTOR: ING. OROZCO BRITO LUIS MSc.

SANGOLQUÍ - ECUADOR

2015

1 CERTIFICADO DEL DIRECTOR DE TESIS



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE POSICIÓN PARA EL CUADRICOPTERO AR.DRONE 2.0 DE PARROT”, realizado por los señores: Germmy Franklin Coloma Vallejo y Kid Coleman Oyos Piarpuesan, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores Germmy Franklin Coloma Vallejo y Kid Coleman Oyos Piarpuesan, para que lo sustente públicamente.

Sangolquí, 30 de Octubre de 2015


Ing. Luis Orozco Brito MSc.
DIRECTOR

2 AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA AUTOMATIZACIÓN Y CONTROL

AUTORIA DE RESPONSABILIDAD

Nosotros: Germmy Franklin Coloma Vallejo con cedula de identidad 1718867987, y Kid Coleman Oyos Piarpuesan con cedula de identidad 1716769334, declaro que este trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE POSICIÓN PARA EL CUADRICOPTERO AR.DRONE 2.0 DE PARROT” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas. Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 30 de Octubre de 2015.


Sr. Germmy Franklin Coloma Vallejo.


Sr. Kit Coleman Oyos Piarpuesan.

3 AUTORIZACIÓN

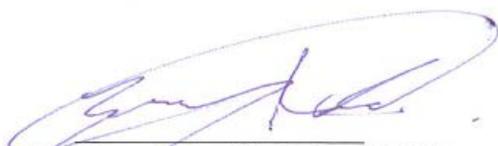


DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE INGENIERIA EN ELECTRÓNICA AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Nosotros: Germmy Franklin Coloma Vallejo y Kid Coleman Oyos Piarpuesan, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE POSICION PARA EL CUADRICOPTERO AR.DRONE 2.0 DE PARROT” cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 30 de Octubre de 2015.



Sr. Germmy Franklin Coloma Vallejo.



Sr. Kit Coleman Oyos Piarpuesan.

DEDICATORIA

A mis padres por ayudarme a cumplir mis objetivos como persona y estudiante, por sus sabios consejos y sacrificios que han hecho, por guiarme y ayudarme a conseguir mis metas, por brindarme los recursos necesarios durante toda la carrera.

A mis hermanos/as por estar siempre presentes, apoyándome y acompañándome para poderme realizar como persona y alentarme en los momentos difíciles.

A todos aquellos que de una u otra manera han creído en mí y me han ayudado a formar la persona que soy.

Germmy Coloma

Dedico este trabajo a las personas más importantes en mi vida, mis padres, que supieron brindarme toda su confianza su apoyo su calidez pero sobretodo estuvieron presentes siempre en los momentos críticos.

A mi hermano que con su autoridad me ha enseñado que todo lo gratificante se tiene que dar un gran esfuerzo por conseguirlo.

Dedicado para mi hermana, este humilde trabajo es una analogía de la realización de las metas que uno se traza en la vida, espero que sea una inspiración para ti, eres la mejor

Coleman Oyos

AGRADECIMIENTO

A mis padres porque siempre han creído en mí, por su apoyo incondicional, por haber inculcado en mí valores y cualidades, y así como por darme la posibilidad de tener una excelente educación.

A mis hermanos/as por estar siempre apoyándome en los momentos de necesidad, por creer en mí y darme el esfuerzo para seguir esta carrera.

A Evelyn por su disposición para brindarme siempre ayuda y animo cuando más lo he necesitado, compartir momentos importantes en mi vida y brindarme todo su amor.

A mi director de tesis el Ing. Luis Orozco por ayudarme a hacer posible este proyecto gracias a su conocimiento y paciencia, a mi codirector el Ing. Darwin Alulema quien colaboró generosamente en la elaboración del proyecto, gracias por el apoyo brindado.

A mis compañeros de universidad y con todas aquellas personas que creyeron en mí y me apoyaron en los altibajos y durante cada momento de mi carrera universitaria que en algún momento formaron parte de mi vida.

Germmy Coloma

A mi padre, mi madre, mi hermana, y mi hermano....mil gracias por todo su apoyo los quiero mucho.

Agradezco a Cinthya Giomar, persona muy especial que llego a mi vida, me enseñó el valor del tiempo y sobretodo la paciencia, gracias por ser maravillosa.

A mi ex jefa Gioconda y mi amigo de labor Galo, gracias por su enseñanza por darme la oportunidad de crecer de manera profesional y brindarme su confianza.

A mis tutores, MSc. Orozco y MSc, Alulema, sus consejos y guías me ha demostrado que la educación nunca se termina, un profesional en la ingeniería debe estar en una constante evolución académica.

Coleman Oyos

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	i
AUTORIZACIÓN.....	iii
RESUMEN	xvii
CAPÍTULO 1.....	1
INTRODUCCIÓN.....	1
1.1 Antecedentes	1
1.2 Justificación.....	2
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos específicos.....	4
1.4 Alcance.....	4
CAPÍTULO 2.....	6
GENERALIDADES.....	6
2.1 Introducción al AR.Drone 2.0.....	6
2.1.1 Componentes mecánicos	9
2.1.1 Componentes electrónicos y de control.....	11
2.1.2 Funcionamiento del cuadricóptero	17
2.1.3 Estudio de los movimientos de los rotores.....	18
2.2 Alternativas de lenguaje de programación	21
2.3 Pruebas de enlace	22
2.4 Variables del cuadricóptero AR.Drone 2.0	23
2.4.1 Variables manipuladas	23
2.4.2 Variables controladas	24
2.4.3 Acceso a las variables del cuadricóptero.....	24
CAPÍTULO 3.....	28
MODELADO DEL CUADRICÓPTERO.....	28
3.1 Modelamiento	28
3.1.1 Metodología para la obtención del modelo	28
3.2 Planificación de experimentos para toma de datos	29

3.3	Estrategia de modelado	38
3.4	Identificación de los parámetros del modelo.....	39
3.4.1	Modelo para la posición `x`	39
3.4.2	Modelo para la posición `y`	47
CAPÍTULO 4.....		55
DISEÑO DE CONTROL DE POSICIÓN DEL CUADRICÓPTERO.....		55
4.1	Sistema de control	55
4.1.1.1	Objetivos del control	55
4.1.1.2	Estrategias de control	56
4.1.1	Componentes del sistema	57
4.1.2	Controlador.....	58
4.1.3	Simulación del sistema de control de posición en Matlab	69
4.1.4	Modificaciones del sistema de control	72
CAPÍTULO 5.....		75
IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DEL CUADRICÓPTERO.....		75
5.1	Estudio y análisis de la interfaz humano/máquina	75
5.2	Programación del sistema de control de posición	88
CAPÍTULO 6.....		96
PRUEBAS Y RESULTADOS		96
6.1	Ensayos y pruebas realizados	96
6.2	Ajustes del sistema.....	108
CAPÍTULO 7.....		131
CONCLUSIONES Y RECOMENDACIONES		131
7.1	Conclusiones	131
7.2	Recomendaciones.....	134
BIBLIOGRAFÍA.....		136
GLOSARIO		138

ÍNDICE DE TABLAS

Tabla 1: Valores característicos de alimentación de voltaje para el AR.Drone 2.0.....	12
Tabla 2: Datos experimentales de entrada para el software TrackDrone Lite 2.0.....	26
Tabla 3: Datos experimentales de salida proporcionados por el software TrackDrone Lite 2.0.....	27
Tabla 4: Efectos de las constantes PID en un sistema en lazo cerrado.....	59
Tabla 5: Parámetros de ajuste del método de oscilación por Ziegles–Nichols.....	62
Tabla 6: Respuesta en el tiempo para las posiciones de los ejes (x, y).....	71
Tabla 7: Comparación de respuesta en el tiempo posición del eje x.....	74
Tabla 8: Comparación de respuesta en el tiempo posición del eje y.....	74
Tabla 9. Descripción de los parámetros y variables que intervienen en el bloque de ingres...	93
Tabla 10. Descripción de los parámetros que intervienen en el bloque de lazo abierto.....	94
Tabla 11. Descripción de las variables que intervienen en el bloque de lazo cerrado.....	95
Tabla 12. Parámetros técnicos y geográficos del lugar de experimentación.....	97
Tabla 13. Características del experimento EXP1.....	99
Tabla 14. Respuesta del sistema de control EXP1.....	103
Tabla 15. Características del experimento EXP2.....	103
Tabla 16. Respuesta del sistema de control EXP2.....	106
Tabla 17. Parámetros técnicos y geográficos del lugar de experimentación final.....	109
Tabla 18. Características del experimento POS1.....	111
Tabla 19. Respuesta del sistema de control POS1.....	114
Tabla 20. Características del experimento POS2.....	114
Tabla 21. Respuesta del sistema de control POS2.....	117
Tabla 22. Características del experimento POS3.....	117
Tabla 23. Respuesta del sistema de control POS3.....	120
Tabla 24. Características del experimento POS4.....	120
Tabla 25. Respuesta del sistema de control POS4.....	123
Tabla 26. Características del experimento POS5.....	123
Tabla 27. Respuesta del sistema de control POS5.....	126
Tabla 28. Características del experimento POS6.....	126
Tabla 29. Respuesta del sistema de control POS6.....	129

ÍNDICE DE FIGURAS

Figura 1a. Comunicación vía control remoto del experimento de Tesla, Nueva York.....	6
Figura 1b. Esquema del primer prototipo UAV, desarrollado por Nikola Tesla.....	7
Figura 2a. Comparación VGA vs QVGA.....	13
Figura 2b. Ángulo de visión.....	13
Figura 3: Esquema del sensor ultrasonido.....	14
Figura 4. Esquema interno general del motor sin escobillas, configuración interna y Configuración eléctrica.....	17
Figura 5. Sistema de movimiento espacial del AR.Drone 2.0.....	18
Figura 6. Movimientos de los rotores.....	19
Figura 7. Movimiento Roll.....	19
Figura 8. Movimiento Pitch.....	20
Figura 9. Movimiento Yaw.....	20
Figura 10. Elevación desde un punto.....	20
Figura 11. Enlace del AR.Drone 2.0 con el computador.....	22
Figura 12. Conexión de red inalámbrica del AR.Drone 2.0.....	22
Figura 13. Conexión de red inalámbrica IP del AR.Drone 2.0.....	23
Figura 14a. Comunicación desactivado.....	23
Figura 14b. Comunicación activado.....	23
Figura 15. Variables que interactúan en el sistema.....	24
Figura 16. Flujo de información entre el AR.Drone 2.0 y el computador.....	25
Figura 17. Entrada y almacenamiento de salidas de los parámetros.....	25
Figura 18. Variables de entrada pitch y roll Archivo Input.txt.....	26
Figura 19. Metodología para la obtención del modelo.....	29
Figura 20. Entrada Pitch y Roll con un ángulo comprendida en 0.5.....	30
Figura 21. Vista frontal del plano de vuelo del cuadricóptero.....	31
Figura 22. Ingreso de variables manipuladas, y salidas de variables a controlar.....	32
Figura 23a. Trayectoria descrita del cuadricóptero a 2600msnm, con ángulo pitch.....	32
Figura 23b. Trayectoria descrita del cuadricóptero a 2600msnm, con ángulo roll.....	33
Figura 24a. Trayectoria descrita del cuadricóptero a 2200msnm, con ángulo pitch.....	34
Figura 24b. Trayectoria descrita del cuadricóptero a 2200msnm, con ángulo roll.....	34
Figura 25a. Trayectoria descrita del cuadricóptero a 1600msnm, con ángulo pitch.....	35

Figura 25b. Trayectoria descrita del cuadricóptero a 1600msnm, con ángulo roll.....	35
Figura 26a. Trayectoria descrita del cuadricóptero a 1100msnm, con ángulo pitch.....	36
Figura 26b. Trayectoria descrita del cuadricóptero a 1100msnm, con ángulo roll.....	36
Figura 27a. Trayectoria descrita del cuadricóptero a 500msnm, con ángulo pitch.....	37
Figura 27b. Trayectoria descrita del cuadricóptero a 500msnm, con ángulo roll.....	37
Figura 28. Planta Posición x con respecto al ángulo Pitch.....	38
Figura 29. Planta Posición y con respecto al ángulo Roll.....	38
Figura 30a. Entrada y salida del sistema Posición del eje x.....	39
Figura 30b. Eliminación tiempo muerto Posición del eje x.....	39
Figura 31a. Estructura del modelo acoplando 1 polo, posición x	40
Figura 31b. Estimación Modelo de primer orden, posición x , calculado por Matlab.....	40
Figura 32a. Estructura del modelo acoplando 1 polo, 1 cero posición x	40
Figura 32b. Estimación Modelo de primer orden, 1 polo, 1 cero posición x	40
Figura 33a. Estructura del modelo acoplando 1 polo, 1 cero y 1 integrador posición x	41
Figura 33b. Modelo de primer orden, 1 polo, 1 cero y 1 integrador posición x	41
Figura 34a. Estructura del modelo acoplando 2 polos, posición x	41
Figura 34b. Estimación Modelo de segundo orden, posición x	41
Figura 35a. Estructura del modelo acoplando 2 polos, 1 cero posición x	42
Figura 35b. Estimación Modelo de segundo orden, 2 polos, 1 cero posición x	42
Figura 36a. Estructura del modelo acoplando 2 polos, 1 cero y 1 integrador posición x	42
Figura 36b. Modelo de segundo orden, 2 polos, 1 cero y 1 integrador posición x	42
Figura 37a. Estructura del modelo acoplando 3 polo, posición x	43
Figura 37b. Estimación Modelo de tercer orden, posición x , calculado por Matlab.....	43
Figura 38a. Estructura del modelo acoplando 3 polos, 1 cero posición x	43
Figura 38b. Estimación Modelo de tercer orden, 3 polos, 1 cero posición x	43
Figura 39a. Estructura del modelo acoplando 3 polos, 1 cero y 1 integrador posición x	44
Figura 39b. Modelo de tercer orden, 3 polos, 1 cero y 1 integrador posición x	44
Figura 40a. Modelo ARX111, posición x	45
Figura 40b. Modelo Estimado ARX 111, posición x	45
Figura 41a. Modelo ARX121, posición x	45
Figura 41b. Modelo estimado ARX 121, posición x	45
Figura 42a. Modelo de la ecuación general de cuarto orden estimado posición x	46
Figura 42b. Función de transferencia identificada posición x en tiempo continuo.....	47

Figura 43a. Entrada y salida del sistema Posición del eje y.....	47
Figura 43b. Eliminación tiempo muerto Posición del eje y.....	47
Figura 44a. Estructura del modelo acoplando 1 polo, posición `y`	48
Figura 44b. Estimación Modelo de primer orden, posición `y`, calculado por Matlab.....	48
Figura 45a. Estructura del modelo acoplando 1 polo, 1 cero posición `y`	48
Figura 45b. Estimación Modelo de primer orden, 1 polo, 1 cero posición `y`	48
Figura 46a. Estructura del modelo acoplando 1 polo, 1 cero y 1 integrador posición `y`	49
Figura 46b. Modelo de primer orden, 1 polo, 1 cero y 1 integrador posición `y`	49
Figura 47a. Estructura del modelo acoplando 2 polos, posición `y`	49
Figura 47b. Estimación Modelo de segundo orden, posición `y`	49
Figura 48a. Estructura del modelo acoplando 2 polos, 1 cero posición `y`	50
Figura 48b. Estimación Modelo de segundo orden, 2 polos, 1 cero posición `y`	50
Figura 49a. Estructura del modelo acoplando 2 polos, 1 cero y 1 integrador posición `y`	50
Figura 49b. Modelo de segundo orden, 2 polos, 1 cero y 1 integrador posición `y`	50
Figura 50a. Estructura del modelo acoplando 3 polo, posición `y`	51
Figura 50b. Estimación Modelo de tercer orden, posición `y`, calculado por Matlab.....	51
Figura 51a. Estructura del modelo acoplando 3 polos, 1 cero posición `y`	51
Figura 51b. Estimación Modelo de tercer orden, 3 polos, 1 cero posición `y`	51
Figura 52a. Estructura del modelo acoplando 3 polos, 1 cero y 1 integrador posición `y`	52
Figura 52b. Modelo de tercer orden, 3 polos, 1 cero y 1 integrador posición `y`	52
Figura 53a. Modelo ARX111, posición `y`	52
Figura 53b. Modelo Estimado ARX 111, posición `y`	52
Figura 54a. Modelo ARX121, posición `y`	52
Figura 54b. Modelo estimado ARX 121, posición `y`	53
Figura 54c. Modelo de la ecuación general de cuarto orden estimado posición `y`	53
Figura 54d. Función de transferencia identificada posición `y` en tiempo continuo.....	54
Figura 55a. Control en lazo cerrado posición `x`	56
Figura 55b. Control en lazo cerrado posición `y`	56
Figura 56. Componentes del cuadricóptero.....	57
Figura 57. Configuración genérica de un controlador PID.....	60
Figura 58. Diagrama del sistema de control en lazo cerrado.....	60
Figura 59. Control PID para el posicionamiento del eje x.....	66

Figura 60. Control PID para el posicionamiento del eje y.....	68
Figura 61. Respuesta simulada con una consigna de 2 para el eje x.....	69
Figura 62. Respuesta simulada con una consigna de -2 para el eje x.....	70
Figura 63. Respuesta simulada con una consigna de 2 para el eje y.....	70
Figura 64. Respuesta simulada con una consigna de -2 para el eje y.....	71
Figura 65. Control PID parámetros autoajustados para el posicionamiento del eje x.....	72
Figura 66. Parámetros autoajustados con una consigna de 2 para el eje x.....	73
Figura 67. Control PID parámetros autoajustados para el posicionamiento del eje y.....	73
Figura 68. Parámetros autoajustados con una consigna de 2 para el eje y.....	74
Figura 69. Entorno TrackDrone Lite 2.0.....	76
Figura 70. Áreas de aplicación del programa.....	77
Figura 71. Dirección IP representado en TrackDrone Lite 2.0.....	78
Figura 72. Tiempo de muestreo e indicador óptimo/fallo de lazo de control.....	78
Figura 73. Botones indicativos para la configuración de condiciones iniciales.....	79
Figura 74. Botones destinados para la acción de vuelo del cuadricóptero.....	80
Figura 75a. Botón de elección modo manual y modo automático.....	80
Figura 75b. Mando de aceleración para modo manual.....	80
Figura 76. Botón para la elección de modos UAV control en lazo abierto.....	81
Figura 77. Botón para la elección de modo UAV para el control en lazo cerrado.....	81
Figura 78. Entorno de instrumentos de navegación.....	82
Figura 79. Zona de configuración de control.....	83
Figura 80. Sector de ingreso de datos, archivo creado para datos Roll y Pitch.....	84
Figura 81. Ingreso de trayectoria, archivo para crear trayectorias.....	84
Figura 82. Sector de almacenamiento de datos de vuelo, archivo generado.....	85
Figura 83. Ingreso de variables de control.....	85
Figura 84. Zona que carga del archivo .txt parámetros de control.....	86
Figura 85. Zona de visualización de desplazamientos del cuadricóptero.....	86
Figura 86. Entorno de desplazamientos del cuadricóptero.....	87
Figura 87. Diagrama de casos de usos aplicado a la metodología Usuario- Sistema-Cuadricóptero.....	88
Figura 88. Implementación del diseño de control.....	89
Figura 89. Archivo con extensión .dll.....	90
Figura 90. Lugar y dirección en donde va archivado "ControlDLL.dll".....	90

Figura 91. Diagrama UML de actividades de la algoritmia de programación.....	91
Figura 92. Diagrama UML de clases.....	92
Figura 93. Bloque de dirección programada del TrackDrone Lite 2.0.....	93
Figura 94. Esquema de programación con ley de control PID.....	94
Figura 95. Esquema del programa realimentado para la acción de control PID.....	95
Figura 96. Planificación de experimentación del sistema de control del cuadricóptero.....	96
Figura 97. Mapa de coordenadas para las pruebas del control de posición EXP1, 2 del cuadricóptero.....	99
Figura 98. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado EXP1.....	100
Figura 99. Respuesta en el tiempo del eje y, EXP1.....	101
Figura 100. Respuesta en el tiempo del eje x, EXP1.....	102
Figura 101. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado EXP.....	103
Figura 102. Respuesta en el tiempo del eje y, EXP2.....	104
Figura 103. Respuesta en el tiempo del eje x, EXP2.....	105
Figura 104. Mapa de coordenadas para las pruebas POS del control de posición del cuadricóptero.....	110
Figura 105. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado POS1.....	111
Figura 106. Respuesta en el tiempo del eje y, POS1.....	112
Figura 107. Respuesta en el tiempo del eje x, POS1.....	113
Figura 108. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado POS2.....	114
Figura 109. Respuesta en el tiempo del eje y, POS2.....	115
Figura 110. Respuesta en el tiempo del eje x, POS2.....	116
Figura 111. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado POS3.....	118
Figura 112. Respuesta en el tiempo del eje y, POS3.....	119
Figura 113. Respuesta en el tiempo del eje x, POS3.....	119
Figura 114. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado POS4.....	121
Figura 115. Respuesta en el tiempo del eje y, POS4.....	122
Figura 116. Respuesta en el tiempo del eje x, POS4.....	122

Figura 117. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado POS5.....	124
Figura 118. Respuesta en el tiempo del eje y, POS5.....	125
Figura 119. Respuesta en el tiempo del eje x, POS5.....	125
Figura 120. Desplazamiento en Matlab del cuadricóptero hacia el punto final deseado POS6.....	127
Figura 121. Respuesta en el tiempo del eje y, POS6.....	128
Figura 122. Respuesta en el tiempo del eje x, POS6.....	128

ÍNDICE DE FOTOGRAFÍAS

Fotografía 1. Proyecto SIVA, creado por el Instituto Nacional de Técnica Aeroespacial, España.....	8
Fotografía 2. Cuadricóptero AR.Drone 2.0 de Parrot.....	9
Fotografía 3. Configuración forma X del AR.Drone 2.0.....	10
Fotografía 4.a. Para lugares abiertos.....	11
Fotografía 4.b. Para lugares cerrados.....	11
Fotografía 5. Batería tipo LIPO.....	12
Fotografía 6. Cámara acoplada al AR.Drone 2.0.....	13
Fotografía 7. Sensor ultrasonido y Placa de navegación.....	15
Fotografía 8. Unidad de medición inercial IMU.....	15
Fotografía 9. Motor inrunner utilizado en el cuadricóptero.....	16
Fotografía 10. Área de experimentación del sistema de control del cuadricóptero.....	98
Fotografía 11. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento EXP1.....	100
Fotografía 12. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento EXP2.....	104
Fotografía 13. Área para realizar el control de posición.....	110
Fotografía 14. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento POS1.....	112
Fotografía 15. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento POS2.....	115
Fotografía 16. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento POS3.....	118
Fotografía 17. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento POS4.....	121
Fotografía 18. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento POS5.....	124
Fotografía 19. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento POS6.....	127

ÍNDICE DE GRÁFICAS

Gráfica 1. Exactitud de la posición para los experimentos EXP1 y EXP2.....	107
Gráfica 2. Precisión de la posición para los experimentos EXP1 y EXP2.....	108
Gráfica 3. Exactitud de la posición en cada uno de los experimentos realizados.....	129
Gráfica 4. Precisión de la posición en cada uno de los experimentos realizados.....	130

RESUMEN

En este trabajo se presenta el desarrollo y validación experimental de un control de posición de un cuadricóptero, se presenta un modelo matemático que describe la dinámica del desplazamiento en vuelo desde un punto inicial hacia un punto final deseado, considerando los aspectos constructivos y la dinámica de los actuadores. Para esto, se utiliza el cuadricóptero AR.Drone 2.0 de la familia Parrot, este tipo de prototipo permite hacer un recambio muy simple de todas y cada una de sus piezas por su estructura que es sencilla y totalmente modular, en el cual se aplica el uso de diferentes herramientas matemáticas para la descripción y orientación de la aeronave, comparando sus ventajas y desventajas ante diversas situaciones de operación. Luego se prosigue por la identificación de los parámetros del cuadricóptero, realizando los ensayos correspondientes en diferentes bancos de pruebas y analizando los resultados obtenidos. A continuación se procede a la síntesis de los controladores a utilizar, optando por sintetizar un control PID, Finalmente se simula el sistema en Matlab, se analizan los resultados de las pruebas hechas en la implementación del prototipo, obteniendo conclusiones acerca del funcionamiento del sistema real.

Palabras Clave: Control de posición, AR.Drone 2.0, Identificación, UAV, Ziegles Nichols, Routh Hurwitz.

CAPÍTULO 1

INTRODUCCIÓN

1.1 Antecedentes

El centro de investigación CICTE¹ está ejecutando el proyecto de reconocimiento, seguridad y video vigilancia con aeronaves no tripuladas, con este propósito, da lugar la necesidad de realizar el trabajo para hacer el modelado, control y la posterior programación de un tipo específico de UAV².

En el Ecuador la Secretaría Técnica de Asentamientos Humanos Irregulares, incorporó la tecnología de vehículos aéreos no tripulados para realizar video vigilancia desde el aire, con el objetivo de prevenir asentamientos ilegales, en el área del Monte Sinaí (noroeste de Guayaquil) (Telegrafo, 2013). Con estos antecedentes en el presente proyecto se plantea el desarrollo de un estudio e investigación, sustentando la apertura de nuevas competencias en el campo de las aeronaves no tripuladas. Dentro de la gama de vehículos no tripulados, se ha elegido como pieza de investigación un cuadricóptero. Un Cuadricóptero o multirotor (ya que tienen más de dos rotores) es un vehículo de vuelo que utiliza cuatro rotores estratégicamente ubicados, con el fin de establecer equilibrio y estabilidad en su eje de rotación, de esta manera permite que el manejo de vuelo sea más factible en relación de maniobrabilidad por parte del usuario. A través de un dispositivo remoto, el usuario envía señales para dirigir el vuelo del cuadricóptero. Hoy en día según los avances de la ciencia, estos dispositivos de mando tales

¹ CICTE Centro de Investigaciones Científicas y Tecnológicas del Ejército.

² UAV Vehículo aéreo no tripulado por sus siglas en inglés Unmanned Aerial Vehicle.

CAPÍTULO 1

como controles remotos han sido sustituidos paulatinamente por dispositivos inteligentes con periferia del estilo táctil, de esta manera la interacción humano/máquina ha sido cada vez más notable y/o amigable, estos vehículos no tripulados, cuentan con un dispositivo de captura de imágenes en tiempo real, con la cual el usuario establece la sensación de tripularlo sin que se llegue a hacerlo, esto a su vez es una innovación ya que esta herramienta permite que en situaciones de riesgo el usuario/piloto del cuadricóptero no sufra ninguna situación de peligro.

En cuanto a plataforma de software para la aplicación de vuelo del cuadricóptero, se utilizaba sistema operativo de Apple (Apple, Apple, 2013), para la exclusividad de la tecnología iPod/iPhone/iPad (Apple, Ipod/Iphone/Ipad, 2013). Sin embargo los desarrolladores han encontrado una estructura sólida en los sistemas operativos de código abierto, como es Android (Android, 2013) enfocado para terminales inteligentes y los sistemas operativos Linux (Linux, 2013) y Windows (Microsoft, 2013) para computadores personales, es así que se presentó una nueva versión de cuadricóptero, y gracias a su versatilidad en proyectos de investigación que acoge este vehículo de vuelo, su código de programación puede ser ejecutado y manipulado para diversos estudios enfocados al desarrollo académico.

1.2 Justificación

El trabajo presenta técnicas de modelado para dar soporte al proyecto de investigación del centro de investigación CICTE, para la ejecución de operaciones de reconocimiento, seguridad y video vigilancia con aeronaves no tripuladas. Por tal motivo la utilización del cuadricóptero AR.DRONE 2.0 de la familia Parrot (PARROT, 2010), como parte de este proyecto se justifica por su bajo costo, sus características aerodinámicas y técnicas que

CAPÍTULO 1

permiten ensayos exigentes de vuelo, su sistema de visión embebido y sus alternativas de programación, lo convierten en una alternativa adecuada para este tipo de proyectos.

Este tipo de modelo permite hacer un recambio muy simple de todas y cada una de sus piezas por su estructura que es sencilla y totalmente modular, permite controlar sus acciones mediante una PC con conexión inalámbrica y hacer una aplicación de control.

Este proyecto es importante porque aporta a la investigación y a la iniciación científica en el Ecuador, ya que se propone el uso de nuevas tecnologías, de la misma manera adquiere importancia debido a que desarrollará una interfaz HMI³, que puede ser ejecutada mediante un computador personal y a las aportaciones en el modelado del sistema de control para monitorizar de forma elemental y poder controlar la posición, sin importar la trayectoria que realice el cuadricóptero.

1.3 Objetivos

1.3.1 Objetivo General

- ❖ Diseñar e implementar un sistema de control de posición para el cuadricóptero AR.DRONE 2.0 de PARROT.

³ HMI Interfaz humano/máquina por sus siglas en ingles Human Machine interface.

1.3.2 Objetivos específicos

- ❖ Realizar el estudio del arte, para definir la plataforma de programación adecuada, que permita implementar el modelado y el control del cuadricóptero AR.DRONE 2.0.
- ❖ Modelar el cuadricóptero utilizando técnicas de identificación paramétrica.
- ❖ Diseñar el sistema de control de posición en el plano (x, y) , del cuadricóptero.
- ❖ Simular el sistema de control de posición del cuadricóptero.
- ❖ Analizar los resultados obtenidos de la simulación para hacer modificaciones al sistema de control diseñado.
- ❖ Estudiar y analizar un HMI que implemente el modelo y el control de posición diseñados.
- ❖ Ejecutar un conjunto de ensayos programados para constatar el cumplimiento de los objetivos de control.

1.4 Alcance

La meta de este proyecto es desarrollar e implementar un programa flexible para controlar la posición, sin importar la trayectoria que realice el cuadricóptero AR.DRONE 2.0 de la familia Parrot (PARROT, 2010).

Con este propósito se realiza el estudio del arte, para poder definir la plataforma de programación adecuada para la implementación del modelado y el control de posición del cuadricóptero.

El sistema a implementarse, pretende modelar el cuadricóptero utilizando metodologías de identificación paramétrica basados en una minimización del error, a pesar que se conoce que

CAPÍTULO 1

el AR.DRONE 2.0 es una planta no lineal, se prevé utilizar técnicas de control conocidas para regular la posición del cuadricóptero, sin importar la trayectoria.

Se desarrolla simulaciones en Matlab (Mathworks, 2013), como parte del diseño del sistema de control de posición de la planta. Posteriormente se prevé analizar los resultados para modificar y/o ajustar el control del cuadricóptero.

Se analiza una interfaz HMI para el monitoreo de variables relacionadas con la posición y la interacción del operador, así como para la implementación del modelo y el control diseñado, a través de la plataforma seleccionada tanto en software como en hardware. Se integrará en un computador personal, el diseño del control realizado en el cuadricóptero requerido para el manejo del usuario/piloto.

Finalmente se plantea ejecutar un conjunto de ensayos para constatar el cumplimiento de los objetivos de control.

En cuanto a la comunicación entre los dispositivos de vuelo y de control, se tendrá que realizar el estudio para definir el uso del canal de comunicación apropiado.

CAPÍTULO 2

GENERALIDADES

2.1 Introducción al AR.Drone 2.0

La historia de los vehículos no tripulados se origina aproximadamente en el año de 1898, dentro del marco de la exhibición eléctrica realizada en Nueva York, el ingeniero inventor Nikola Tesla (York, 1998), realiza la transmisión de movimientos vía control remoto hacia un bote a menor escala dentro de una piscina, con una consola a distancia. Dando como resultado que el bote realice maniobras de acuerdo a las exigencias dadas, logrando el origen de las transmisiones inalámbricas como se observa en la Figura 1a. En la Figura 1b se observa la incursión de prototipos dirigidos inalámbricamente, proponen nuevas invenciones para el desarrollo de sistemas de vehículos autónomos.

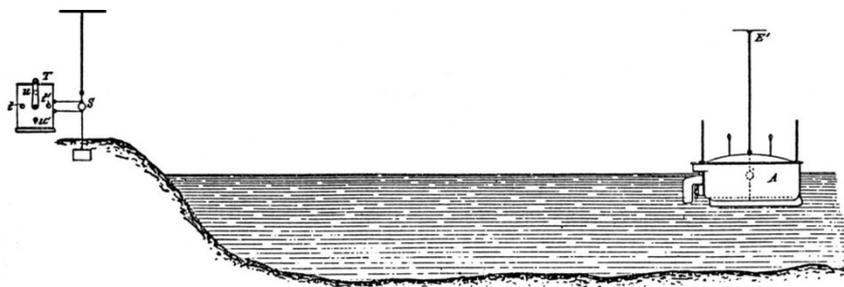


Figura 1a. Comunicación vía control remoto del experimento de Tesla, Nueva York, 1898.

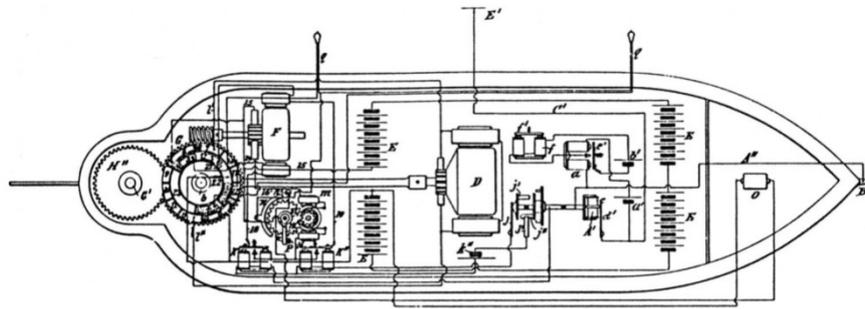


Figura 1b. Esquema del primer prototipo UAV, desarrollado por Nikola Tesla, vista superior.

Es así que la relación de pilotos a distancia con vehículos de cualquier característica y movilidad, presenta una evolución y/o revolución tecnológica a medida que los requisitos lo exijan; tanto en el área de vigilancia, investigación, búsqueda, exploración y hasta en el ámbito militar.

En especial interés, los vehículos aéreos no tripulados UAV's⁴ sirven para facilitar labores de reconocimiento en áreas de difícil acceso, este antecedente implica que su parte inteligente presente un control estable y constante mediante modelos matemáticos. El control PID es aquel modelo que más comúnmente se halla en las computadoras de los vehículos de vuelo, ya que permiten mayor control para sistemas tridimensionales.

Las exigencias que tienen los UAV's en la actualidad, hace que su estructura física tenga diferencias significativas en varios modelos para su utilidad, existen UAV's del tipo aeroplano que desarrollan velocidades muy altas. Por citar ejemplo el proyecto SIVA (INTA, 2006), originario de España, es un vehículo para sistemas tácticos de vigilancia tanto para el orden militar como civil, desarrollado por el INTA⁵, este UAV alcanza velocidades máximas

⁴ UAV's Vehículos aéreos no tripulados por sus siglas en inglés Unmanned Aerial Vehicle

⁵ INTA Instituto Nacional de Técnica Aeroespacial, España

CAPÍTULO 2

de hasta 190km/h, con un peso aproximado de 300Kg, y como tiempo de autonomía de 7 horas como se ilustra en la Fotografía 1.



Fotografía 1. Proyecto SIVA, creado por el Instituto Nacional de Técnica Aeroespacial, España, tomado de (INTA, 2006).

En comparación con los UAV's anteriormente analizados, existen vehículos que no requiere desarrollar altas velocidades, con características en mejor estabilidad en un punto geográfico. Estos UAV's son multirrotores, ya que su sistema de vuelo se sustenta en el acoplamiento de varios rotores en su estructura física.

En base a ello, hay diferentes UAV's en relación al número de rotores y mejor estabilidad en el aire, un ejemplo a citar es el cuadricóptero que se ilustra en la Fotografía 2, una estructura mecánica de cuatro brazos en forma de cruz que tiene acoplado 4 rotores estratégicamente ubicados en el extremo de cada brazo, con la finalidad de obtener una acción de vuelo y maniobra más constante y firme, en algunos modelos viene acoplado un sistema de captura de video en tiempo real para ofrecer mejor interacción con el sistema de vuelo del cuadricóptero y el piloto.

La mecánica del cuadricóptero en comparación con los helicópteros convencionales es más versátil y dinámica, ya sea en su diseño y mantenimiento, por citar un punto importante el cambio de las palas del rotor del helicóptero requiere de conexiones mecánicas para lograrlo,

CAPÍTULO 2

mientras que en el cuadricóptero no es necesario. Al poseer cuatro rotores el cuadricóptero tiene la capacidad de volar con menos energía cinética obteniendo como resultado el reducir daños en algún rotor, mientras que el helicóptero no puede sustentar dicha ventaja y logra ser menos funcional.

Para propósitos de investigación el cuadricóptero AR.Drone del fabricante Parrot, es muy versátil en cuanto a programación, robustez y de igual manera en su costo ya que es accesible obteniendo prestancias excelentes.

El cuadricóptero AR.Drone 2.0; es una estructura de ingeniería móvil para vuelo no tripulado, con las características de maniobrabilidad y mejor estabilidad en el aire. Este vehículo móvil presenta propiedades aerodinámicas de mejor desempeño de vuelo, gracias a la posición de los rotores que están alojados estratégicamente en su estructura, y que son controlados por un microcontrolador.



Fotografía 2. Cuadricóptero AR.Drone 2.0 de Parrot, tomado de (PARROT, 2010)

2.1.1 Componentes mecánicos

Las partes mecánicas del cuadricóptero, son esencialmente robustas y resistentes para condiciones de vuelo en circunstancias de dificultad moderada. El AR.Drone 2.0 es una

CAPÍTULO 2

estructura modular que contempla diversos componentes y que son fabricados por materiales a prueba de golpes, hay que indicar que estas partes contemplan diversas especificaciones y análisis que se detalla a continuación.

El cuerpo es la parte en donde se encuentra el circuito electrónico de la computadora, que recibe los comandos a distancia del control remoto, y que permite la acción de los movimientos para el vuelo.

También esta acoplado la fuente de alimentación, el cual provee de energía a los rotores y la cámara de video en tiempo real. Hay que destacar que tiene un puerto USB, esto con la finalidad de obtener las grabaciones de todo el ciclo de vuelo que ha recorrido el AR.Drone 2.0.

Las extremidades conforman el soporte para los rotores y paletas que conlleva a la acción de vuelo, hechas con fibras de carbono, este material presenta rigidez y ligereza al mismo tiempo, y conlleva que los rotores obtengan la capacidad de desarrollar la potencia necesaria para elevar todo el peso del AR.Drone. Su configuración física viene en forma X como se ilustra en la Fotografía 3.



Fotografía 3. Cuerpo o parte central, configuración forma X del AR.Drone 2.0

CAPÍTULO 2

Este modelo tiene dos corazas las cuales están fabricados con espuma de polipropileno y se enganchan con suma facilidad al cuerpo del vehículo mediante un sistema de imanes, la primera coraza es ligera y protege el cuerpo del AR.Drone 2.0 para exteriores como se ilustra en la Fotografía 4a, y es concebida con el objetivo de volar en espacios abiertos al aire libre. Mientras que la segunda coraza es destinada para vuelos en lugares cerrados, esta estructura aparte de dar protección al cuerpo, también da seguridad a los rotores y paletas juntos mediante circunferencias acopladas a la coraza como se muestra en la Fotografía 4b, detalle primordial que hace que tenga más durabilidad y robustez.



Fotografía 4a. Para lugares abiertos



Fotografía 4b. Para lugares cerrados

2.1.1 Componentes electrónicos y de control

Para el funcionamiento del AR.Drone 2.0, la fuente de alimentación requiere de una batería tipo LIPO⁶ como se ilustra en la Fotografía 5. Estas baterías usadas comúnmente en el aeromodelismo, tienen la característica de reducir el espacio para que el vehículo sea más compacto y de esta manera que el peso no influya en la acción de vuelo. Sin embargo su tiempo de carga es un inconveniente ya que esta batería tarda aproximadamente una hora en

⁶ LIPO Baterías de polímero de Litio

CAPÍTULO 2

tener el 100% de carga para utilizarla nuevamente. La batería del AR.Drone 2.0 tiene las siguientes características que se muestra en la Tabla 1.



Fotografía 5. Batería tipo LIPO

Tabla 1: Valores característicos de alimentación de voltaje para el AR.Drone 2.0

Batería tipo	LIPO
Corriente nominal	100mAh
Voltaje (100%)	12.5Volts
Voltaje (0%)	9Volts
Autonomía de vuelo	12 a 15 minutos
Tiempo de carga	90 minutos

Fuente: (PARROT, 2010)

Elaborado por: Autores del proyecto

Al disminuir el voltaje a un rango mínimo, el AR.Drone 2.0 envía un mensaje al usuario como medida de advertencia, si existe el caso de no tomar en cuenta tal advertencia, el sistema detectará los niveles bajos de voltaje y automáticamente se apaga, con la salvedad de que el vehículo aterriza para evitar la alimentación nula en el momento de vuelo. Para la captura en video de vuelo del AR.Drone 2.0, viene acoplado una cámara tipo CMOS⁷, que se encuentra adaptada en la parte frontal del vehículo, con el fin de codificar y transmitir el video

⁷ CMOS Semiconductor complementario de óxido metálico por sus siglas en inglés Complementary metal-oxide-semiconductor

CAPÍTULO 2

capturado hacia un terminal, en un dispositivo táctil o una computadora personal como se ilustra en la Fotografía 6.



Fotografía 6. Cámara acoplada al AR.Drone 2.0

Esta cámara soporta formatos QVGA⁸, este formato es usado comúnmente para dispositivos móviles que entregan resoluciones de los 320x240 pixeles. Se denomina QVGA ya que es del tamaño de una cuarta parte del formato VGA⁹ que tiene una resolución de 640x480 pixeles (Figura 2a). El lente de la cámara tiene un ángulo de visión de 90 grados obteniendo una vista panorámica (Figura 2b).

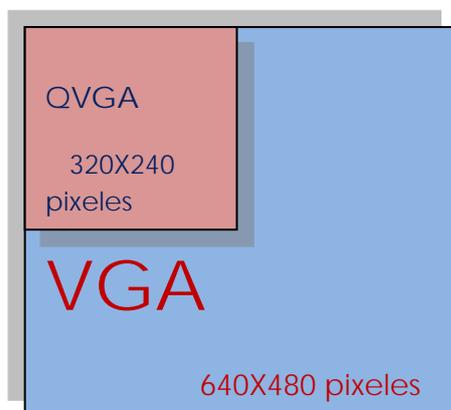


Figura 2a. Comparación VGA vs QVGA

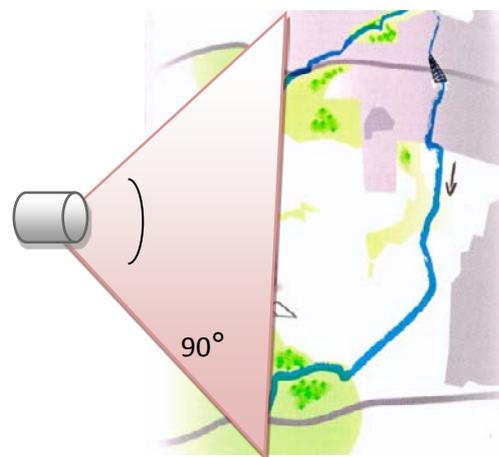


Figura 2b. Ángulo de visión

⁸ QVGA por sus siglas en inglés Quarter Video Graphics Array

⁹ VGA Adaptador gráfico de video por sus siglas en inglés Video Graphics Array

CAPÍTULO 2

El AR.Drone 2.0 tiene instalado sensores que monitorean el correcto funcionamiento del sistema en el instante de vuelo, y que están acoplados por debajo del cuerpo central. De fabricación tipo MEMS¹⁰, tecnología basada en la micro-electromecánica (Maluf & Williams, 2004), estos sensores aportan mediciones en parámetros de altitud, obstáculos en línea de vuelo, como también en control de velocidad vertical en el entorno real.

En la Figura 3 se muestra un sensor de ultrasonidos, el cual permite medir la distancia libre de obstáculos, su funcionamiento describe el comportamiento sonar de los murciélagos con el propósito de no chocar con objetos en la oscuridad. En el plano técnico, el sensor emite un pulso ultrasónico, si existe una barrera dicho pulso se refleja y se transforma en señal eléctrica, otro pulso consecuentemente al anterior se emite cuando el eco dejado por el primer pulso desaparezca.

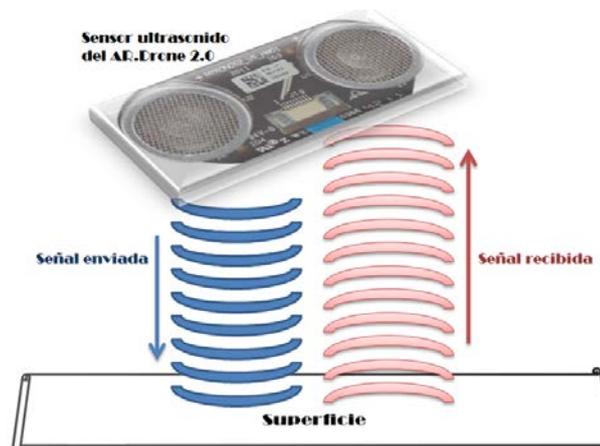


Figura 3. Esquema del sensor ultrasónico

La placa de navegación para el AR.Drone 2.0 está equipada con una serie de sensores para asegurar la estabilidad del cuadricóptero. Esta placa también cuenta con un sensor ultrasónico,

¹⁰ MEMS Sistemas micro electromecánicos por sus siglas en inglés Micro Electro Mechanical Systems

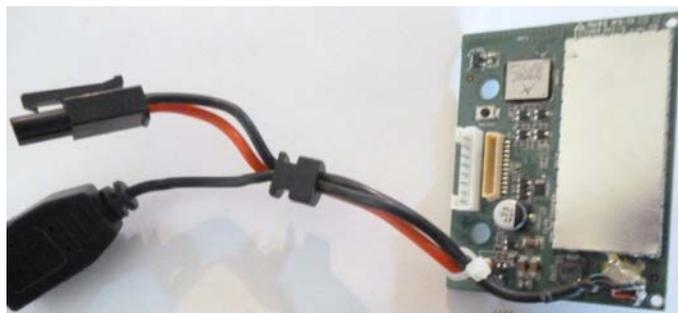
CAPÍTULO 2

un altímetro, un acelerómetro, un giroscopio de dos ejes, un giroscopio piezoeléctrico de precisión y un magnetómetro como se aprecia en la Fotografía 7.



Fotografía 7. Sensor ultrasonido y Placa de navegación

Tiene una unidad de medición inercial IMU¹¹ como se ilustra en la Fotografía 8, el cual es una serie de acelerómetros digital de 3 ejes, monitorea los movimientos de posición. Un giroscopio de 3 ejes, esta unidad es utilizada para monitorear y estabilizar automáticamente los parámetros de movimiento lineal y de velocidad angular, un GPS¹² que puede ser incorporado externamente en la unidad que establece posición en un lugar determinado en el instante de vuelo, un magnetómetro de ejes el cual sirve para la orientación del cuadricóptero.



Fotografía 8. Unidad de medición inercial IMU

¹¹ IMU Unidad de medición inercial por sus siglas en inglés Inertial Measurement Unit

¹² GPS Sistema de posicionamiento global por sus siglas en inglés Global Positioning System

CAPÍTULO 2

Los motores que utiliza el cuadricóptero son tipo sin escobilla 'inrunner', también denominados brushless, estos son utilizados en la actualidad en el campo aeroespacial, equipos de automatización, ventilación y/o instrumentación. La característica de estos motores brushless no emplean escobillas en la conmutación, la conmutación se lo realiza de forma electrónica. Estos motores consumen 15 Watts y 29 Watts de potencia en modo Hovering¹³, dando como resultado generar 3.500 rpm. El rango de velocidades de los motores oscila entre 10.300 hasta 41.400 rpm como se aprecia en la Fotografía 9. Para diferenciar la parte trasera de la delantera del AR.Drone 2.0, en la placa de control de cada motor cuentan con led's indicadores rojo y verde. Para el control de los nodos de los motores posee un microcontrolador AVR de baja potencia de 8 bits y un ADC¹⁴ de 10 bits.



Fotografía 9. Motor inrunner utilizado en el cuadricóptero

Gracias a esta característica el proceso de conmutación no genera transferencia de energía, de esta forma no se presenta calor ni rozamiento en las partes internas del motor. De igual manera aumenta el rendimiento y por ende no hace falta realizar el mantenimiento periódico, al contrario de los motores de escobilla que si genera calor y su tiempo de vida disminuye.

¹³ Hovering Modo de vuelo en suspensión

¹⁴ ADC Convertidor análogo digital por sus siglas en inglés Analog Digital Converter

CAPÍTULO 2

El torque de rotación es generado por la interacción entre el campo magnético producido por las bobinas del rotor y el campo magnético del estator que se muestra en la Figura 4.

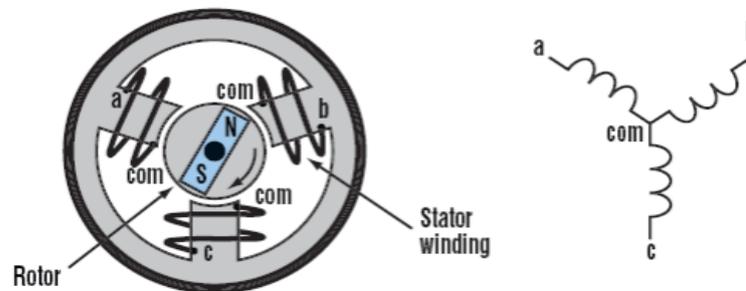


Figura 4. Esquema interno general del motor sin escobillas, configuración interna y configuración eléctrica, tomado de (Perez, 2003).

El campo magnético rotante del estator se produce energizando las bobinas en forma conmutada, de modo que el torque sea siempre máximo. Para energizar las bobinas adecuadas de manera que se produzca el máximo par posible en la posición actual del rotor, es necesario conocer cuál es esa posición, para lo cual se requiere un sensor de posición angular que informa la configuración del rotor en tiempo real. La interacción entre la posición censada, y la lógica de energización de las bobinas se lleva a cabo mediante una lógica de control formando un lazo realimentado de control (Perez, 2003).

2.1.2 Funcionamiento del cuadricóptero

El vehículo tiene 6DOF¹⁵, esto representa que el cuadricóptero se mueva en sistema de coordenadas ortogonal combinado con el sistema de rotación, sus ejes (X, Y, Z, y los ángulos de navegación Roll¹⁶, Pitch¹⁷, Yaw¹⁸), el dispositivo dispone de una unidad inercial de bajo

¹⁵ 6DOF 6 grados de libertad por sus siglas en inglés 6 Degrees Of Freedom

¹⁶ Roll Giro alrededor del eje Xm, denominado alabeo (ϕ)

¹⁷ Pitch Giro alrededor del eje Ym, denominado cabeceo (θ)

¹⁸ Yaw Giro alrededor del eje Zm, denominado guiñada (ψ)

CAPÍTULO 2

coste a partir de la cual se estiman los ángulos de pitch, roll y yaw, asimismo, el algoritmo de estimación que viene incorporado en el cuadricóptero calcula la posición y velocidad referente al eje de referencia situado en el centro de masa del sistema como se detalla en la Figura 5.

En la fase inicial del vuelo, el vehículo despegar, se estabiliza y en ese punto se fija el sistema de coordenadas (punto inicial x, y, z), en ese instante coincidirá con el sistema de coordenadas del cuadricóptero (X_m, Y_m, Z_m), a partir de ese momento empiezan los ensayos, y los sensores suministrarán posiciones y velocidades.

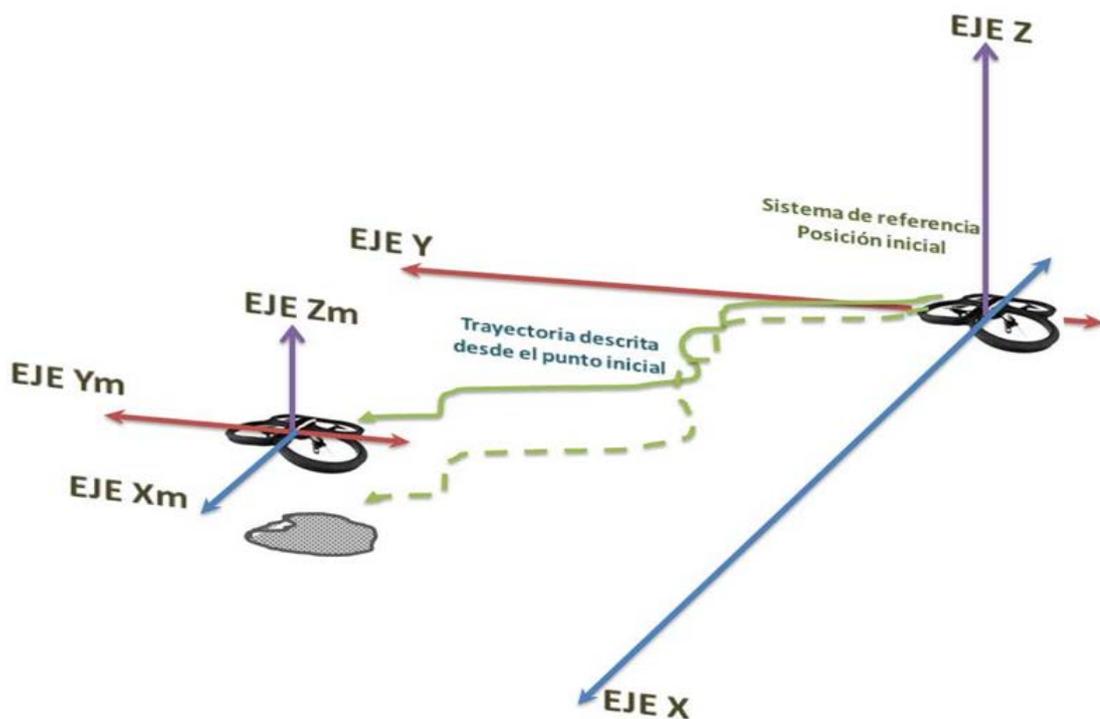


Figura 5. Sistema de movimiento espacial del AR.Drone 2.0

2.1.3 Estudio de los movimientos de los rotores

Para que la estructura mecánica del cuadricóptero no tenga caídas, el control de vuelo se basa en la variación de la velocidad angular de cada uno de los rotores, y para eso es necesario

CAPÍTULO 2

variar la potencia entregada a cada uno de ellos, con el fin de aumentar o disminuir en revoluciones para que genere el torque adecuado.

Cada par de rotores opuestos giran en el mismo sentido (Figura 6), es así que los rotores left y right giran en sentido horario, mientras que los rotores front y rear giran en sentido anti horario, al realizar esta acción de movimiento de los rotores el cuadricóptero estará en una posición de levitación.

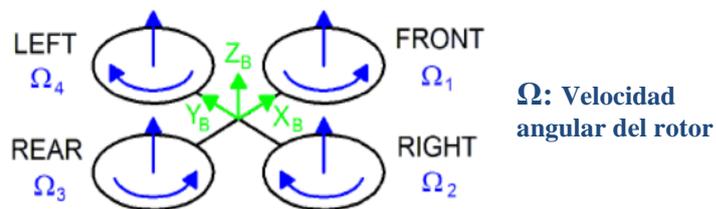


Figura 6. Movimientos de los rotores, tomado de (Piskorski, Brulez, Eline, & D`Haeyer, 2012)

En la Figura 7 se muestra como se realiza el movimiento Roll, el circuito de control incrementa la velocidad angular del rotor left y disminuye en la misma proporción la velocidad angular del rotor right.

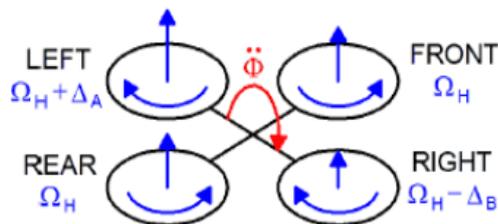


Figura 7. Movimiento Roll, tomado de (Piskorski, Brulez, Eline, & D`Haeyer, 2012)

En la Figura 8 se muestra como se realiza el movimiento Pitch, el circuito de control incrementa la velocidad angular del rotor rear y en la misma proporción disminuye la velocidad angular del rotor front.

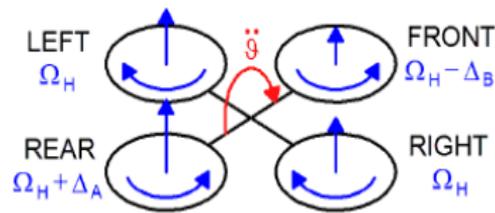


Figura 8. Movimiento Pitch, tomado de (Piskorski, Brulez, Eline, & D`Haeyer, 2012)

El movimiento Yaw se desarrolla incrementando la velocidad angular en el par de rotores left y right, y disminuye la velocidad angular en el par de rotores rear y front, todos en la misma proporción como se ilustra en la Figura 9.

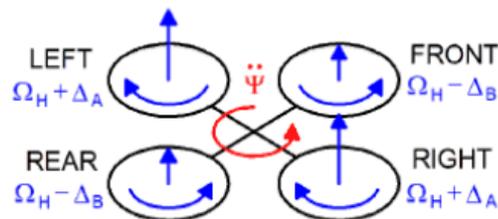


Figura 9. Movimiento Yaw, tomado de (Piskorski, Brulez, Eline, & D`Haeyer, 2012)

Para que el cuadricóptero se eleve, basta con incrementar los valores de la velocidad de todos los rotores, con esto se pretende que el empuje que realiza los rotores hace que el cuadricóptero despegue como se puede observar en la Figura 10.

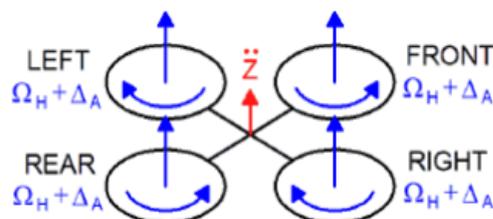


Figura 10. Elevación desde un punto, tomado de (Piskorski, Brulez, Eline, & D`Haeyer, 2012)

2.2 Alternativas de lenguaje de programación

El AR.Drone 2.0 puede ser programado con varias alternativas entre las cuales es C# que es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, su sintaxis básica deriva de C/C++, es similar a Java, aunque incluye mejoras derivadas de otros lenguajes. Su creador Anders Heljsberg fue también el creador de muchos otros lenguajes y entornos como Turbo Pascal, Delphi o Visual J++. Las alternativas con respecto a programas destinados para el enlace, comunicación, programación y posteriormente control del cuadricóptero son indefinidas, muchos de estos programas dependen de las características de las plataformas en la cual están programadas, y también obedece al sistema operativo en el cual son desarrollados, para efecto de la programación del diseño de control de posición del cuadricóptero AR.Drone 2.0 y de forma exclusiva para este proyecto de investigación, se ha tomado como elección representativa al programa TrackDrone Lite 2.0 realizado por la Universidad Politécnica De Valencia (UPV, 2012).

La idea principal detrás del lenguaje es combinar la prestancia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic. La interfaz AR.Drone control.net empleada es de licencia pública general GNU¹⁹, es decir que su código fuente es de libre acceso para el público.

Otra alternativas de lenguaje utilizado para el desarrollo de la aplicación ha sido Java, y el entorno Eclipse. Java ofrece un sencillo tratamiento de las conexiones por puertos UDP y TCP y además la posibilidad de crear una interfaz gráfica en pocos pasos. Como base para el desarrollo de la aplicación de control del AR.Drone.

¹⁹ GNU Sistema operativo libre.

CAPÍTULO 2

2.3 Pruebas de enlace

Cuando el AR.Drone 2.0 está encendido crea un punto de acceso local ad-hoc WIFI²⁰, que debe ser unido al computador para poder realizar cualquier acción de control, como se observa en la Figura 11.



Figura 11. Enlace del AR.Drone 2.0 con el computador

En la Figura 12 se observa la red inalámbrica WIFI que crea automáticamente el servidor AR.Drone 2.0.

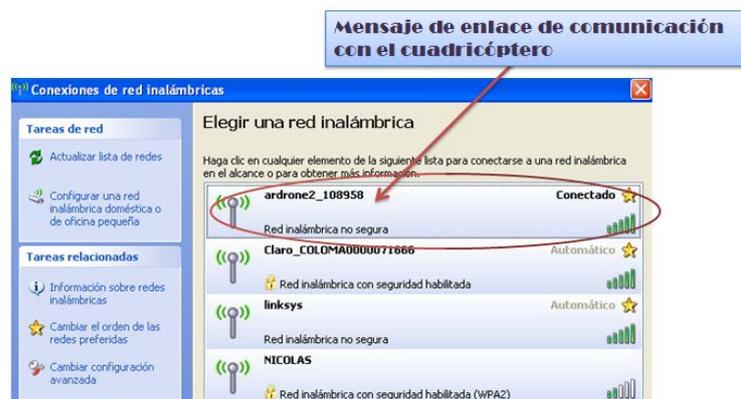


Figura 12. Conexión de red inalámbrica del AR.Drone 2.0

En la Figura 13 se ilustra el estado de conexión de la red inalámbrica mediante una dirección IP²¹ desde el servidor DHCP²² Drone. El ordenador externo puede comenzar a comunicarse

²⁰ WIFI Fidelidad inalámbrica por sus siglas en inglés Wireless Fidelity

²¹ IP Protocolo de internet por sus siglas en inglés Internet Protocol

²² DHCP Protocolo de configuración hosting dinámico por sus siglas en inglés Dynamic Host Configuration Protocol

CAPÍTULO 2

con el avión no tripulado usando la interfaz proporcionada por el fabricante. La interfaz se comunica a través de cuatro canales, cada uno con un puerto UDP²³ diferente.

	Tipo de dirección:	Asignada por DHCP
	Dirección IP:	192.168.1.2
	Máscara de subred:	255.255.255.0
	Puerta de enlace pred.:	192.168.1.1

Figura 13 Conexión de red inalámbrica IP

El AR.Drone 2.0 se comunica con el software de aplicación para la verificación de vuelo, en este caso se usa el TrackDrone Lite 2.0, se pulsa la opción START FLYING como se observa en la Figura 14a, el programa se enlaza automáticamente con el AR.Drone 2.0 enviando un mensaje de comunicación realizada, indicado un mensaje STOP FLYING como se puede observar en la Figura 14b.



Figura 14a. Comunicación desactivada



Figura 14b. Comunicación activada

2.4 Variables del cuadricóptero AR.Drone 2.0

2.4.1 Variables manipuladas

Es la cantidad o condición de la planta que se modifica a fin de influir sobre la variable controlada a través de la dinámica de la planta (Ogata, 1998).

²³ UDP Protocolo de diagrama del usuario por sus siglas en inglés User Datagram Protocol

CAPÍTULO 2

Las variables de los ángulos pitch y roll corresponden a las manipuladas como se observa en la Figura 15.

2.4.2 Variables controladas

Es la cantidad o condición que se mide y controla, normalmente las variables controladas es considerada la salida del sistema.

El cuadricóptero posee tres tipos de movimientos en los ejes (x, y, z), las posiciones (x, y), son considerados como las variables controladas, teniendo como referencias las entradas, como se muestra en la Figura 15.

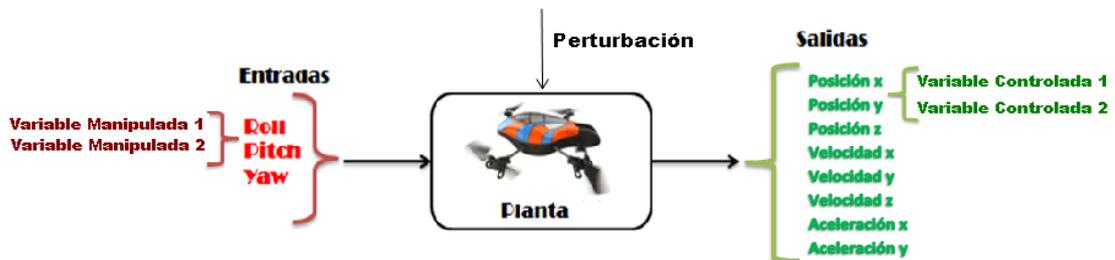


Figura 15. Variables que interactúan en el sistema

2.4.3 Acceso a las variables del cuadricóptero

Para realizar un control óptimo capaz de posicionar al cuadricóptero, compensando cualquier tipo de perturbaciones y mantener estable; es necesario conocer sus parámetros dinámicos experimentales, accediendo a las variables del cuadricóptero.

En la Figura 16 se muestran los principales canales de comunicación, la trama de video y los datos de los sensores son enviados del cuadricóptero hacia el computador y los comandos de control para el movimiento desde la PC hacia el AR.Drone 2.0.

CAPÍTULO 2



Figura 16. Flujo de información entre el AR.Drone 2.0 y el computador.

El comando que genera el desplazamiento del cuadricóptero en el eje Y mediante el movimiento Roll, y el desplazamiento en el eje X mediante el movimiento Pitch es el comando AT*PCMD. En la Figura 17, se muestra como ingresar las variables de entrada pulsando en la opción load input, se carga el archivo Input.txt como se aprecia en la Figura 19, donde están almacenadas los ángulos de referencia pitch y roll, estas variables son a dimensionales y pueden variar en el rango [-1,1], los cuales corresponden a los límites de inclinación máximos y mínimos fijados.



Figura 17. Entrada y almacenamiento de salidas de los parámetros.

La información del sistema debe contener datos significativos de tal manera que se debe considerar ciertas características sobre dicha información, como elección de los correctos datos a ser utilizados, el período de muestreo y la cantidad de datos.

CAPÍTULO 2

La elección de los datos que se va a utilizar se analiza a partir del capítulo III, en la tabla 2 se muestra con más detalle los valores de entrada de referencia.

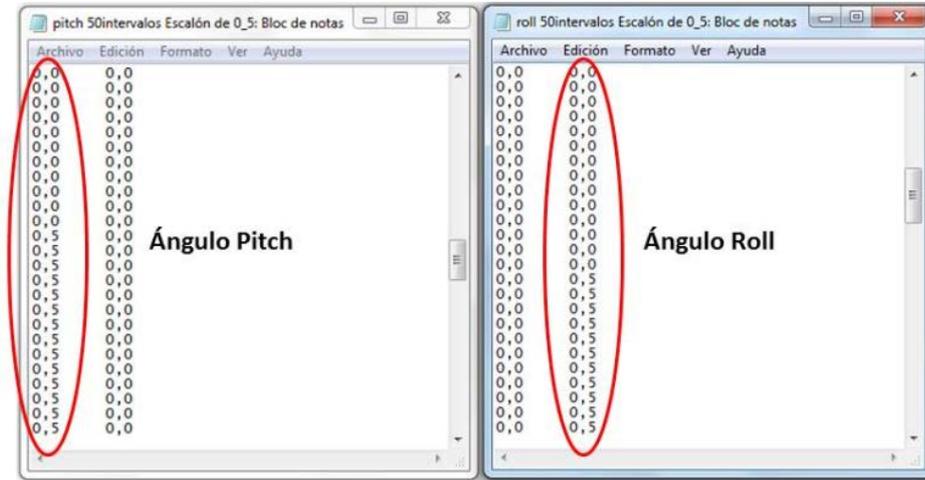


Figura 18. Variables de Entradas Pitch y Roll, Archivo Input.txt.

La información en cada columna para la tabla 2 es:

Columna 1: Ref_Pitch entre -1 y 1

Columna 2: Ref_Roll entre -1 y 1

Tabla 2: Datos experimentales de entrada para el software TrackDrone Lite

Ref_Pitch	Ref_Roll
0.0	0.0
0.0	0.0
0.15	0.0
0.15	0.0
-0.15	0.0
-0.15	0.0
0.0	0.15
0.0	0.15
0.0	-0.15
0.0	-0.15

Elaborado por: Autores del proyecto

CAPÍTULO 2

Se guardan las variables de salida experimentales en el archivo data.txt, mostrados en la tabla 3, de esta manera se puede acceder a las variables del cuadricóptero, las variables vienen por columnas en un fichero de texto que se puede cargar en Matlab para su respectivo modelamiento:

La información en cada columna para la tabla 3 es:

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

Columna 6: Velocidad del eje X en metros/segundos

Columna 7: No hay datos

Columna 8: No hay datos

Columna 9: No hay datos

Tabla 3: Datos experimentales de salida proporcionados por el software TrackDrone Lite 2.0

Entrada Roll	Posición en Y	Velocidad en Y	Entrada Pitch	Posición en X	Velocidad en X	No uso	No uso	No uso
0,0	0,0	0,0	0,0	0,0	0,0	0,00	0,00	0,00
0,0	0,0	0,0	0,0	0,0	0,0	0,00	0,00	0,00
0,0	0,0	0,0	0,0	0,0	0,0	0,00	0,00	0,00
0,0	0,0	0,0	0,0	0,0	0,0	0,00	0,00	0,00
0,0	0,0	0,0	0,0	0,0	0,0	0,00	0,00	0,00
0,5	0,0	0,0	0,5	0,0	0,0	0,00	0,00	0,00
0,5	0,0	0,0	0,5	1,221	0,0	0,00	0,00	0,00
0,5	-3,85	-0,035	0,5	1,221	0,02	0,00	0,00	0,00
0,5	-3,85	-0,041	0,5	1,221	0,023	0,00	0,00	0,00
0,5	-3,85	-0,048	0,5	1,221	0,026	0,00	0,00	0,00
0,0	-3,85	-0,053	0,0	1,221	0,129	0,00	0,00	0,00
0,0	-3,85	-0,058	0,0	1,221	0,131	0,00	0,00	0,00
0,0	-3,85	-0,062	0,0	1,221	0,133	0,00	0,00	0,00

Elaborado por: Autores del proyecto

CAPÍTULO 3

MODELADO DEL CUADRICÓPTERO

3.1 Modelamiento

La estimación del modelo dinámico de movimiento del AR.Drone 2.0, que relaciona las entradas de ángulos con las velocidades y posiciones, se basa en las respuestas de vuelo real obtenidos experimentalmente proporcionados por el software TrackDrone lite 2.0, el estudio del proceso de identificación y modelado del cuadricóptero se lleva a cabo mediante el software Matlab. Para ello, y de acuerdo al tercer objetivo específico planteado en 1.3.2, se escoge como variables manipuladas a los ángulos pitch, roll y como variables controladas, a las posiciones (x, y) , mostradas en la Figura 15.

3.1.1 Metodología para la obtención del modelo

En la Figura 19, se esquematiza la metodología para el modelado, empezando por la toma de datos en este paso se recopila toda la información de las variables experimentales; en el acondicionamiento de datos se elimina ruido, tiempo muerto, y se analiza la respuesta del sistema; se elige la estructura del modelo agregando o eliminando polos, ceros e integradores, el último paso se ajusta el modelo estimando sus parámetros.



Figura 19. Metodología para la obtención del modelo

3.2 Planificación de experimentos para toma de datos

La primera decisión es definir las variables de interés, y qué valores deben ser manipulados para excitar al sistema durante el experimento. Se debe tener en cuenta que pueden existir señales no deseadas que, aunque afecten a la evolución de la salida, no pueden considerarse como variables manipuladas debido a la imposibilidad de actuar sobre ellas. En el caso de que estas señales puedan ser medidas, pueden considerarse también como entradas al sistema, midiéndose sus valores durante el experimento caso contrario, deben ser consideradas como perturbaciones.

La elección del período de muestreo está directamente relacionada con las constantes de tiempo del sistema, y tiene una influencia decisiva en el experimento de identificación. Así, un período de muestreo muy pequeño puede llevar a la obtención de datos redundantes, que no aportan información sobre el sistema (pero sí ocupan espacio en la memoria del dispositivo

CAPÍTULO 3

de almacenamiento de datos), mientras que un período de muestreo demasiado grande provoca grandes dificultades a la hora de identificar la dinámica del sistema, por medio del software TrackDrone Lite 2.0, se consideró un tiempo de muestreo de 60ms, con la finalidad de que el cálculo de procesamiento del control aplicado al cuadricóptero no permita pérdida de datos.

Las entradas al sistema deben ser cuidadosamente elegidas de forma que las respuestas recogidas proporcionen toda la información posible sobre el sistema, para el ingreso de las variables manipuladas se da un escalón ascendente de [0 ,0.5], seguido por un escalón descendente de [0.5 ,0], es necesario ingresar el segundo escalón descendente para que el cuadricóptero no se aleje exageradamente hasta perder comunicación, obteniendo como resultado un pulso como se observa en la Figura 20, con 50 muestras cada intervalo, dando un tiempo de muestreo de 9 segundos, esto se obtiene al multiplicar las muestras totales por el tiempo de muestreo.

Número de muestras $\rightarrow m = 150$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación $\rightarrow t_{exp} = m * t_s \Rightarrow 150 * 0.06$

$$t_{exp} = 9[s]$$

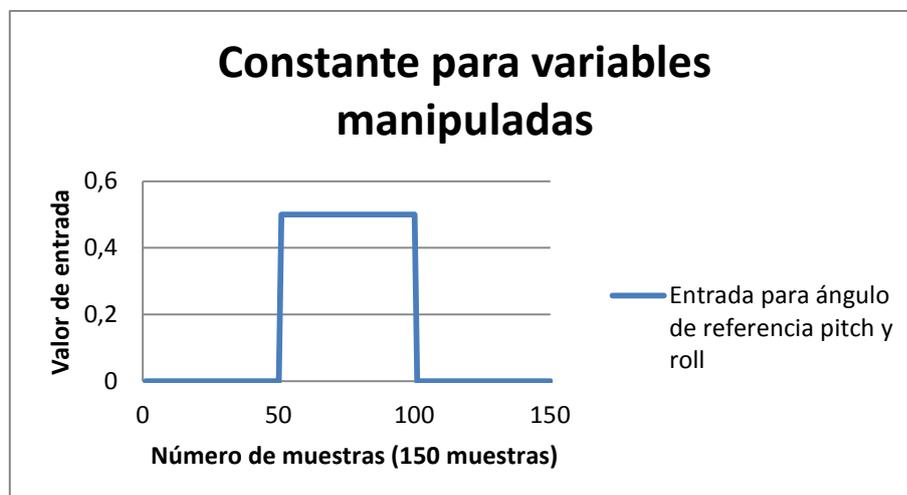


Figura 20. Entrada Pitch y Roll con un ángulo comprendida en 0.5

CAPÍTULO 3

En principio, cuanto más información se tenga sobre el sistema, más exacto será el proceso de identificación, para el AR.Drone 2.0, se prevé realizar los experimentos en el plano cartesiano X-Y, tomando como referencia la dirección de la cámara de video del cuadricóptero con el eje `x` positivo, y de manera consecuente el eje `y` positivo hacia la derecha de la cámara de video del mismo, y a una altura de 1 metro sobre el piso, tal como se ilustra en la Figura 21.

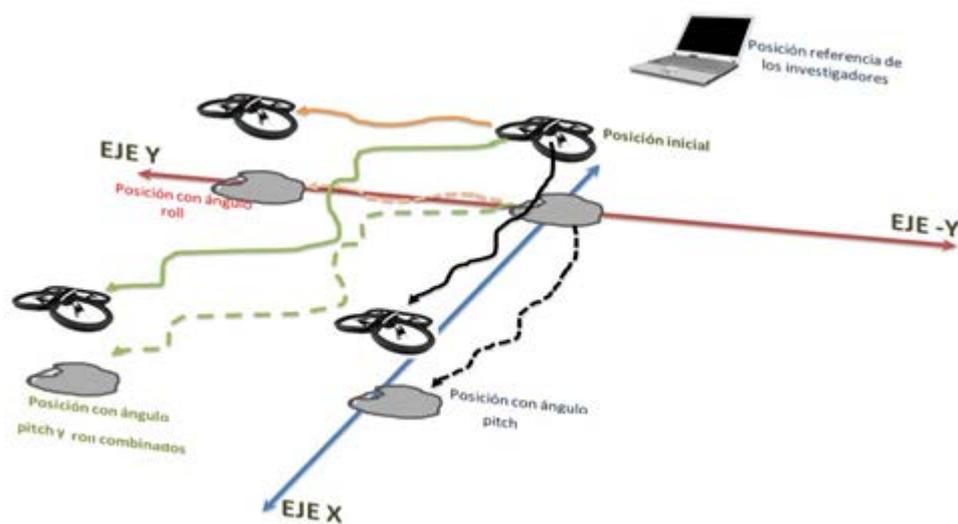


Figura 21. Vista frontal del plano de vuelo del cuadricóptero

En la Figura 21, se aprecia las posiciones que se desea obtener una vez que se ingrese el ángulo pitch, como para el ángulo roll por separado, obteniendo como resultado una posición en `x` o `y`.

Para los primeros experimentos de vuelo con el cuadricóptero se ingresa valores de entrada únicamente para el ángulo pitch, luego de recopilar los datos de navegación que entrega el TrackDrone Lite 2.0, como se muestra en la Figura 22, se lo hará de la misma manera para el caso de ángulo roll, y tomando como lugar de ensayo la localidad de la ciudad de Quito, que tiene una altitud a 2.600 metros sobre el nivel del mar.

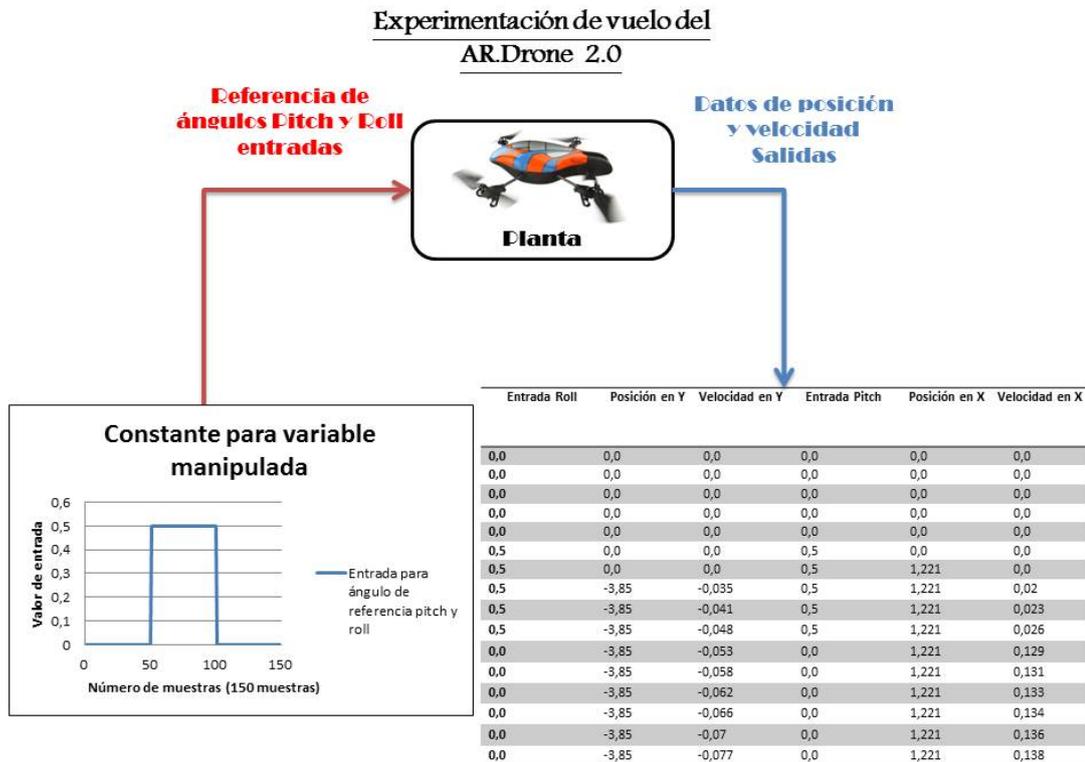


Figura 22. Ingreso de variables manipuladas, y salidas de variables a controlar

Experimentación a 2600m sobre el nivel del mar.

Al ingresar el ángulo pitch y roll como se indica en la Figura 22, se observa que el cuadricóptero no tiene reacción al excitar con un pulso, se ingresa un tren de pulsos para observar su comportamiento como se indica en las Figuras 23a y 23b.

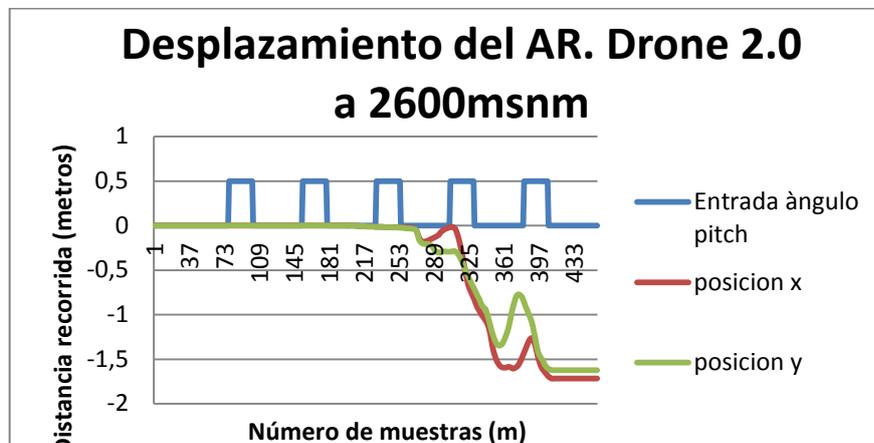


Figura 23a. Trayectoria descrita del cuadricóptero a 2600msnm, con ángulo pitch

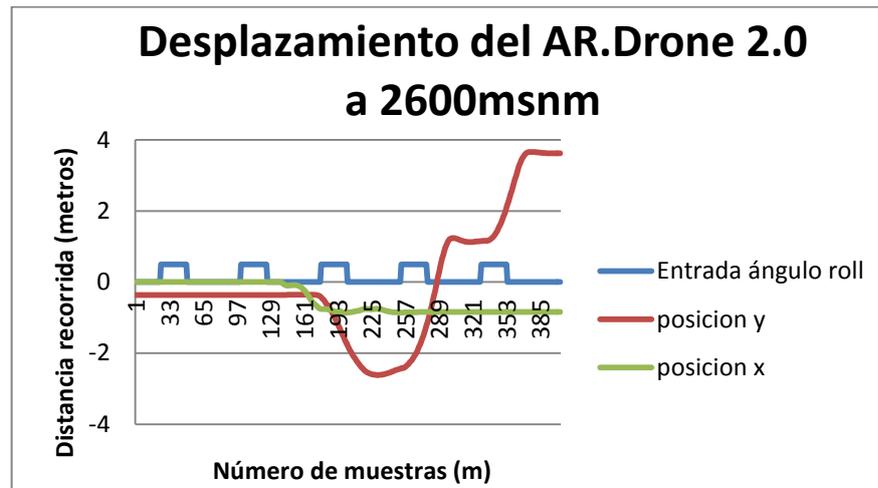


Figura 23b. Trayectoria descrita del cuadricóptero a 2600msnm, con ángulo roll

Se observa en la Figura 23a, que el cuadricóptero presenta una trayectoria errática, y un desplazamiento en el plano (x, y), Figura 23b, con el ángulo roll como excitación se produce diferentes variaciones en la posición (x, y).

El cuadricóptero presenta movimiento errático en el instante de vuelo, ya que no despegue de una manera correcta, al ingresar una variable manipulada se obtiene como resultado variaciones en dos salidas, esto representa un sistema MIMO²⁴. Además existe un retardo puro (tiempo muerto), y variaciones en la posición de `x` e `y`, que pueden provenir de un modelo NO LINEAL y/o con parámetros variantes en el tiempo. Por lo tanto, a criterio de investigación los siguientes ensayos se los realiza a una altitud menor, para obtener posibles comportamientos aproximadamente lineales e invariantes con el tiempo (LTI).

Experimentación a 2200m sobre el nivel del mar.

A esta altitud el comportamiento del cuadricóptero es más eficaz pero sigue presentando movimiento errático, ya que en algunos instantes no presenta el desplazamiento deseado al ingresar los ángulos de pitch o roll, como se muestra en la Figura 24a y 24b.

²⁴ MIMO Sistema con varias entradas y salidas por sus siglas en inglés multiple inputs and multiple outputs

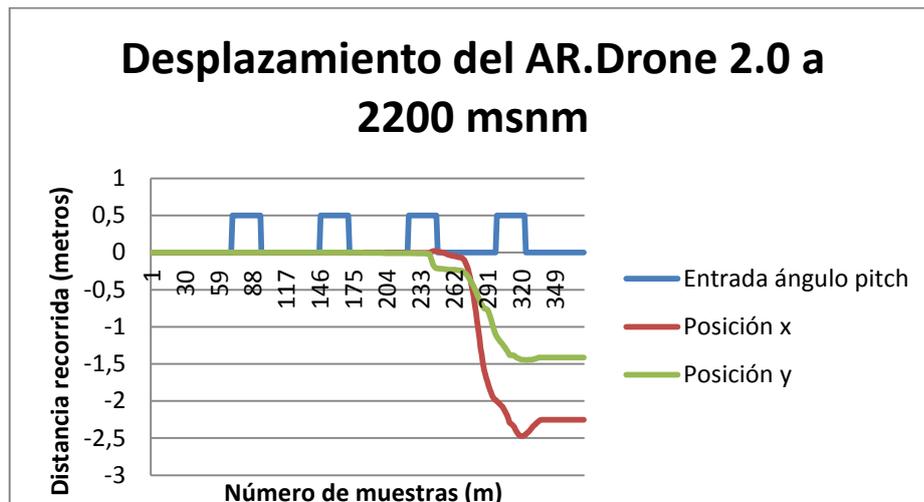


Figura 24a. Trayectoria descrita del cuadricóptero a 2200msnm, con ángulo pitch

Existe un desplazamiento no deseable en la posición Y, el sistema sigue comportándose como un sistema MIMO.

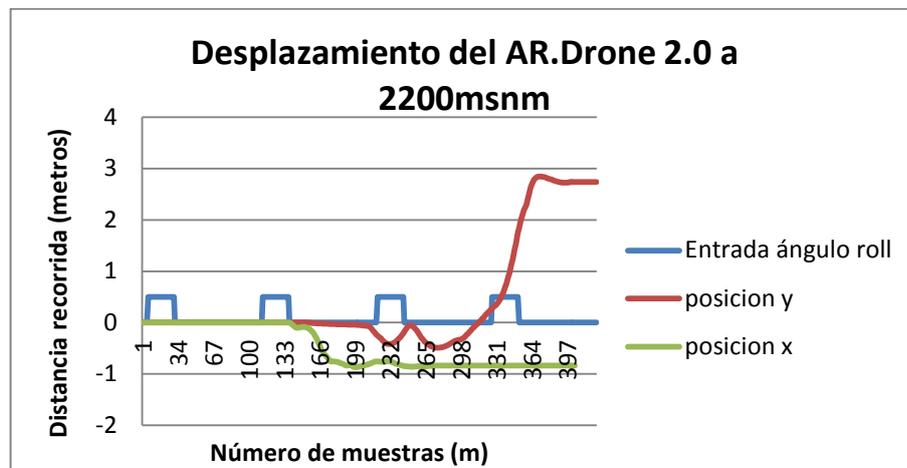


Figura 24b. Trayectoria descrita del cuadricóptero a 2200msnm, con ángulo roll

De igual manera lo mismo que ocurre cuando se ingresa valor de ángulo de referencia pitch, sucedió para el caso de ángulo roll, existe un desplazamiento no deseado en la posición x, que dificulta el proceso de identificación. Se realiza la experimentación para el comportamiento del sistema a una altitud menor a 2200msnm.

Experimentación a 1600m sobre el nivel del mar.

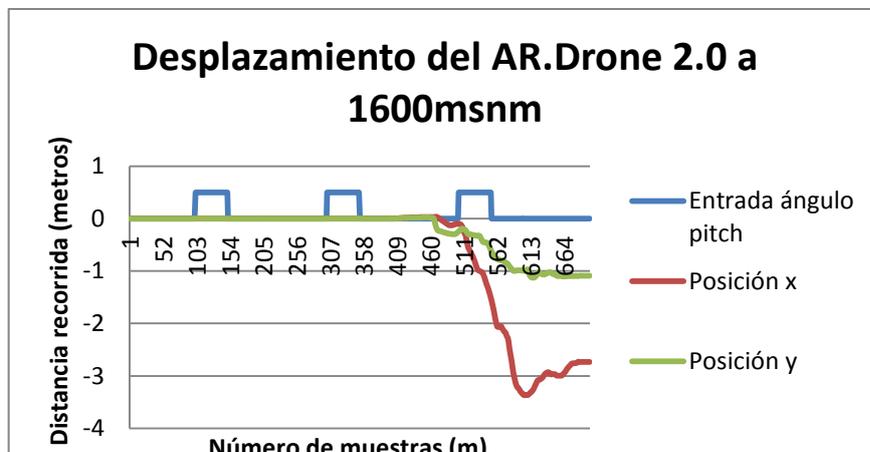


Figura 25a. Trayectoria descrita del cuadricóptero a 1600msnm, con ángulo pitch

De los múltiples experimentos realizados a la altura de 1600msnm se aprecia que existe una disminución en la relación de desplazamientos indeseables para cada uno de los ejes (x, y), que son excitados con las variables manipuladas correspondiente. En la Figura 25a, la posición en eje x es menor en comparación a la medida cuando se realizó la experimentación a una altitud de 2200msn, con estas hipótesis se comienza a deducir que bajando a regiones con menor altitud, la inestabilidad comienzan a disminuir, y se comporta como un sistema aproximado lineal, de igual manera ocurre cuando se ingresa valores para ángulo roll, como se muestra en la Figura 25b.

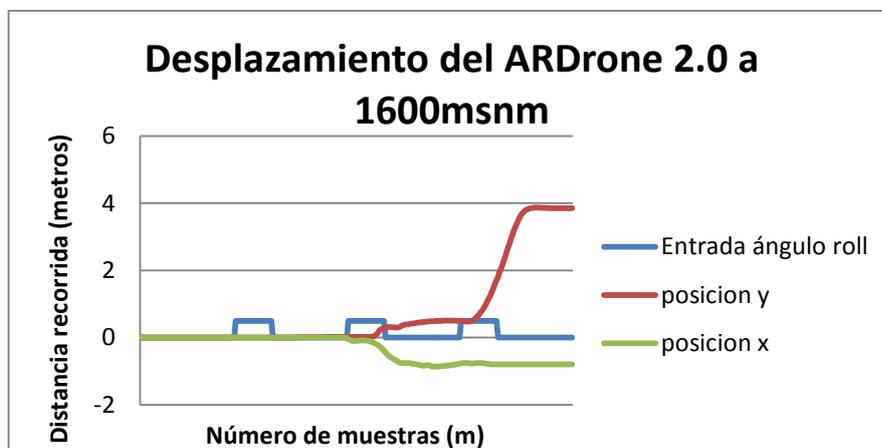


Figura 25b. Trayectoria descrita del cuadricóptero a 1600msnm, con ángulo roll

Experimentación a 1100m sobre el nivel del mar.

Como se ha visto en altitudes desde 1600msnm, y en relación con esta altitud de ensayo, estos desplazamientos indeseables y de igual manera el sub pico (undershoot), para el eje y (Figura 24b), comienzan a disminuir. Las Figuras 26a y 26b, representan las respuestas esperadas, la propuesta es bajar a una altura que comprenda los 200msnm hasta 800msnm, este criterio se lo utiliza con la intención de poder mejorar los movimientos erráticos.

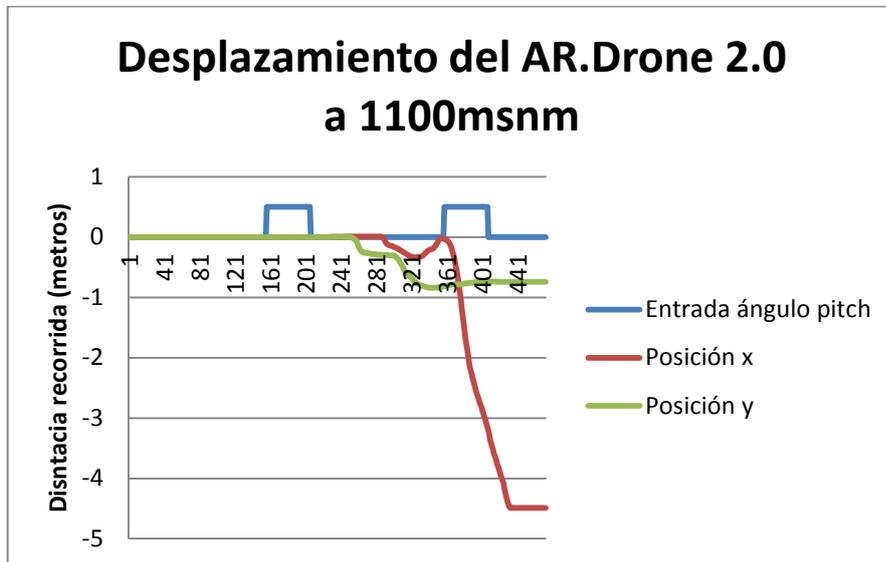


Figura 26a. Trayectoria descrita del cuadricóptero a 1100msnm, con ángulo pitch

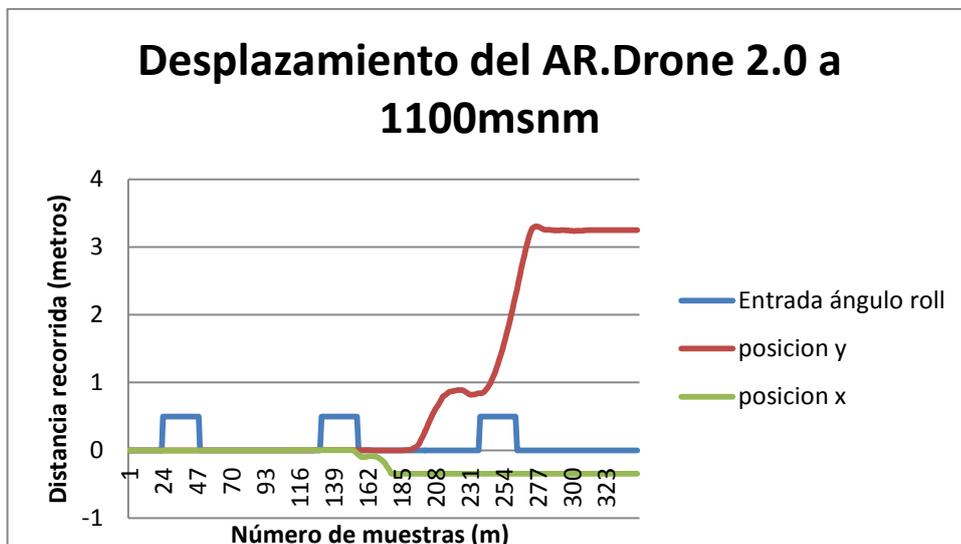


Figura 26b. Trayectoria descrita del cuadricóptero a 1100msnm, con ángulo roll

Experimentación a 500m sobre el nivel del mar.

A una altura de 500 metros sobre el nivel del mar, el cuadricóptero presenta mejores movimientos y precisión de posicionamiento, esto hace que el sistema se comporte como un sistema SISO, no existen posiciones indeseables como se observa en las Figuras 27a y 27b.

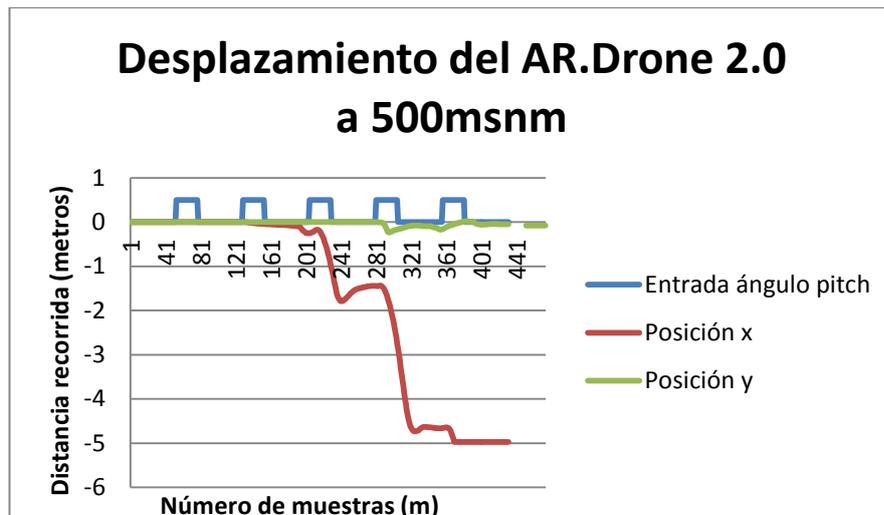


Figura 27a. Trayectoria descrita del cuadricóptero a 500msnm, con ángulo pitch

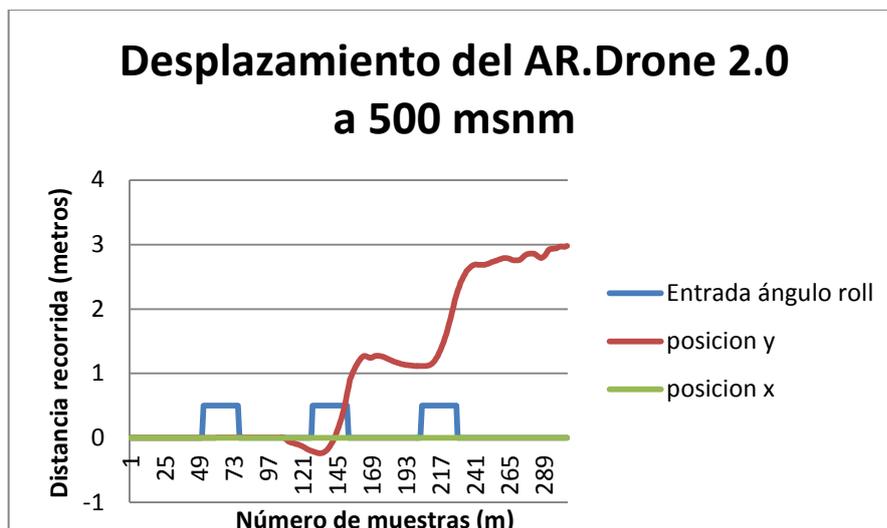


Figura 27b. Trayectoria descrita del cuadricóptero a 500msnm, con ángulo roll

Las respuestas de las posiciones en los ejes (x, y), registrados están generalmente acompañados de ruidos indeseados u otro tipo de imperfecciones que puede ser necesario

CAPÍTULO 3

corregir antes de iniciar la identificación del modelo. Se trata, por tanto, de preparar los datos para facilitar y mejorar el proceso de identificación, se determina que a una altura de 500 metros sobre el nivel del mar, el AR.Drone 2.0 presenta estabilidad, y un comportamiento lineal e invariante con el tiempo.

3.3 Estrategia de modelado

Los cuadricópteros son sistemas multivariantes o MIMO, y no lineales en los que una entrada afecta a varias salidas y recíprocamente una salida es afectada por varias entradas, son más laboriosos de modelar (Figura 15), como se observa su comportamiento en alturas de 2600 a 1100 metros sobre el nivel del mar, es por eso que la estrategia de modelado se lo hace como dos sistemas SISO²⁵ lineal de simple entrada y simple salida, se toma los valores experimentales a 500 metros sobre el nivel del mar (Figuras 27a, 27b), ya que su comportamiento es aproximadamente lineal e invariante con el tiempo, para el ángulo de referencia pitch posición 'x', Figura 28, y el ángulo de referencia roll posición 'y', Figura 29.



Figura 28. Planta Posición 'x' con respecto al ángulo Pitch



Figura 29. Planta Posición 'y' con respecto al ángulo Roll

²⁵ SISO Sistema con una entradas y una salidas por sus siglas en ingles single inputs and single outputs

CAPÍTULO 3

Para enfrentar el problema de estimación del modelo se plantean varias hipótesis basadas en las variables, los modelos para cada eje de movimiento se encuentran desacoplados y definen las siguientes relaciones, ángulo pitch con posición `x`, ángulo roll con posición en `y`.

3.4 Identificación de los parámetros del modelo.

Para este propósito se utiliza como herramienta la identificación de Matlab denominado “ident”, tomando en consideración los siguientes criterios, el sistema se lo modela como un proceso lineal en tiempo continuo representando sus polos y ceros, el grado de la planta obedece exclusivamente a la aproximación más fiel al comportamiento del sistema alterado por los valores de entrada, seguido por implementar un modelo en tiempo discreto ARX²⁶.

3.4.1 Modelo para la posición `x`

En la Figura 30a, muestra las variables de entrada y salida del sistema, se acondiciona los datos eliminando el tiempo muerto para identificar y modelar como se aprecia en la Figura 30b.

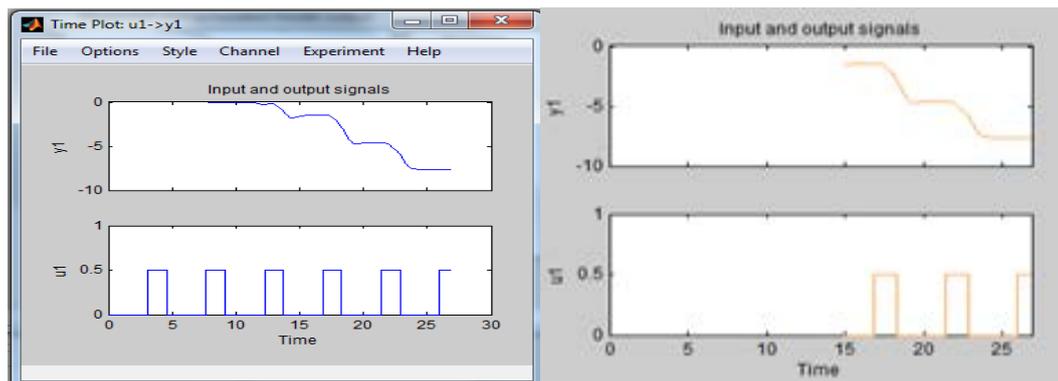


Figura 30a. Entrada y salida del sistema
Posición del eje x

Figura 30b. Eliminación tiempo muerto
Posición del eje x

Se elige la estructura añadiendo 1 polo, $\frac{X(s)}{\theta(s)} = \frac{K}{(1+T_{p1}s)}$, como se la aprecia en la Figura 31a,

mediante la relación del sistema de primer orden con respecto a la planta, esta presenta una

²⁶ ARX Por sus siglas en ingles Auto Regressive with Exogenous input

CAPÍTULO 3

correlación con respecto a los valores experimentales en un -83.41% como se observa en la Figura 31b.

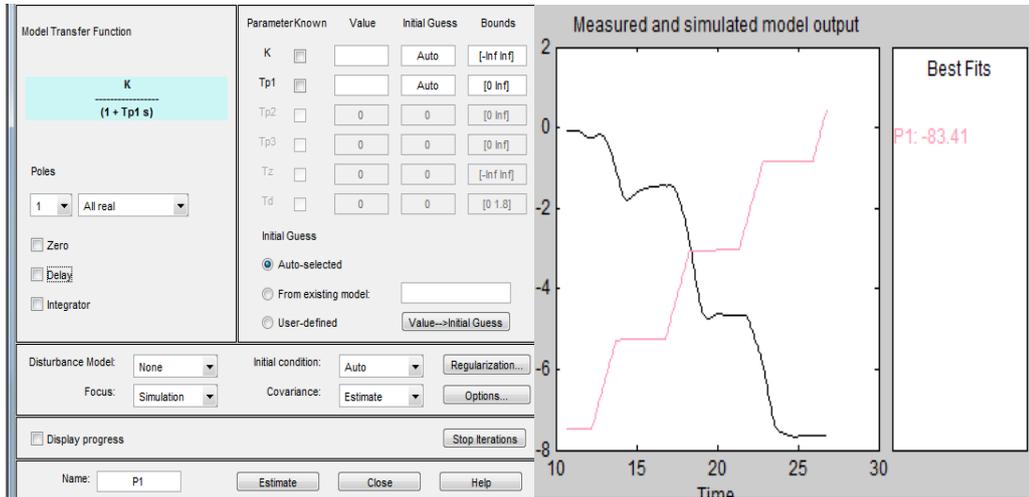


Figura 31a. Estructura del modelo acoplado 1 polo, posición 'x'

Figura 31b. Estimación Modelo de primer orden, posición 'x', calculado por Matlab

Estimación añadiendo 1 polo y un cero, su estructura describe: $\frac{X(s)}{\theta(s)} = \frac{K(1+T_z s)}{(1+T_{p1} s)}$, como se la aprecia en la Figura 32a, mediante la relación del sistemas de primer orden y un cero con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -53.8% como se observa en la Figura 32b.

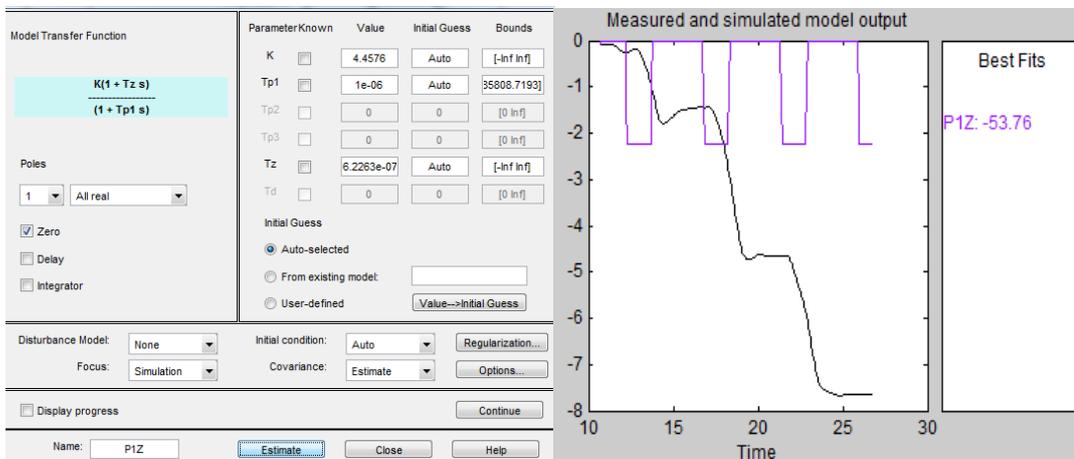


Figura 32a. Estructura del modelo acoplado 1 polo, 1 cero, posición 'x'

Figura 32b. Estimación Modelo de primer orden, 1 polo, 1 cero posición 'x'.

CAPÍTULO 3

Estimación añadiendo 1 polo, 1 cero y un integrador, su estructura describe:

$$\frac{X(s)}{\theta(s)} =$$

$$\frac{K(1+T_z s)}{s(1+T_{p1} s)}$$

como se la aprecia en la Figura 33a, mediante la relación del sistemas de segundo orden con un cero y un integrador con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -53.8% como se observa en la Figura 33b.

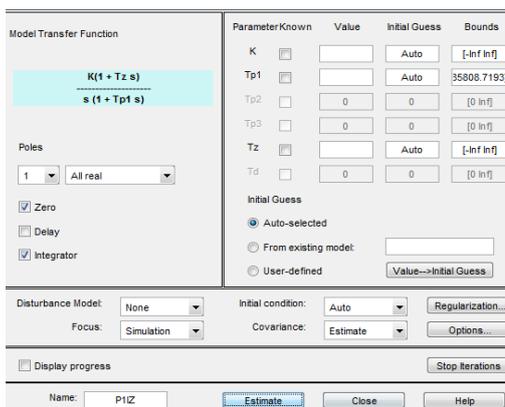


Figura 33a. Estructura del modelo 1 polo, 1 cero y 1 integrador posición 'x'

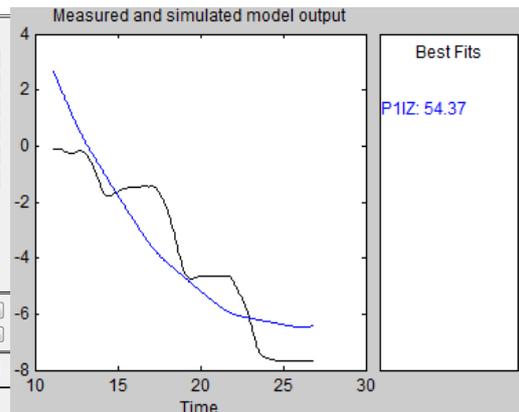


Figura 33b. Estimación de segundo orden 1 polo, 1 cero y 1 integrador posición 'x'

Estimación añadiendo 2 polos, su estructura describe: $\frac{X(s)}{\theta(s)} = \frac{K}{(1+T_{p1} s)(1+T_{p2} s)}$, como se la

aprecia en la Figura 34a, mediante la relación del sistemas de segundo orden con 2 polos, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -54.21% como se observa en la Figura 34b.

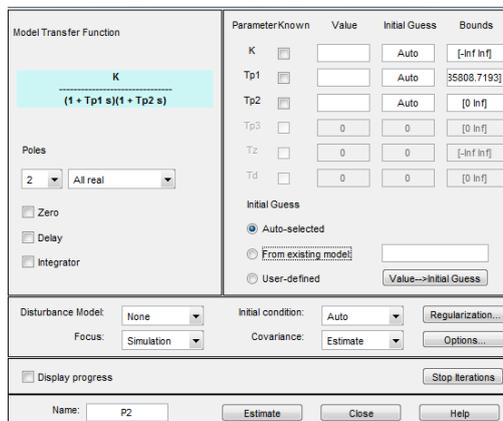


Figura 34a. Estructura del modelo 2 polos, posición 'x'

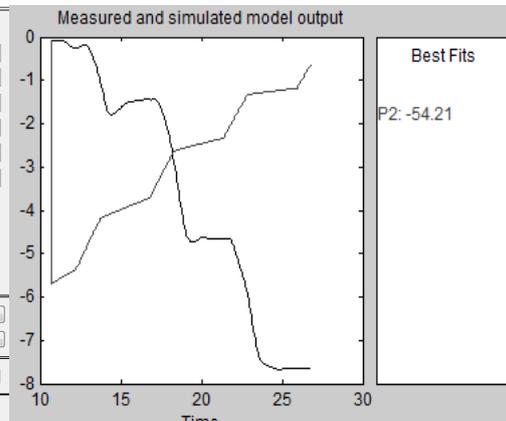


Figura 34b. Estimación de segundo orden 2 polos, posición 'x'

CAPÍTULO 3

Estimación añadiendo 2 polos un cero, su estructura describe: $\frac{X(s)}{\theta(s)} = \frac{K(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}$, como se

la aprecia en la Figura 35a, mediante la relación del sistemas de segundo orden con 2 polos y un cero, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 6.99% Figura 35b.

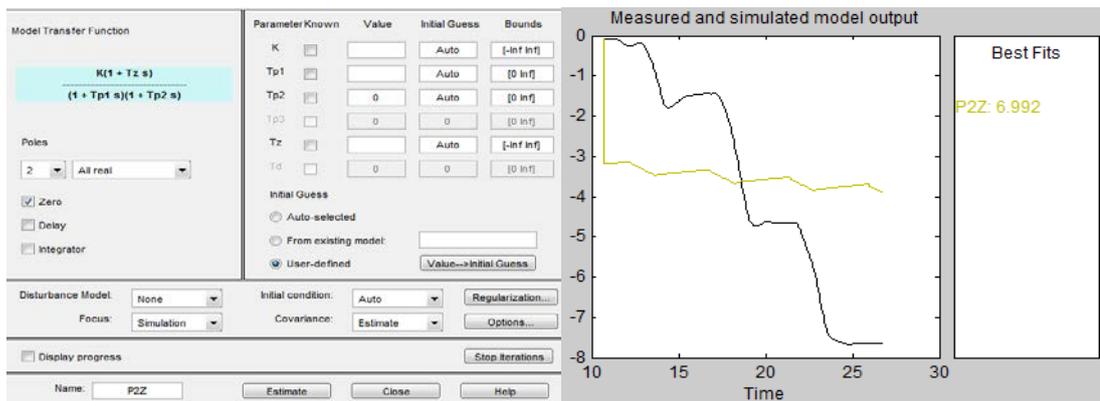


Figura 35a. Estructura del modelo 2 polos, 1 cero, posición `x`

Figura 35b. Estimación de segundo orden, 2 polos, 1 cero, posición `x`

Estimación añadiendo 2 polos un cero y un integrador, su estructura describe: $\frac{X(s)}{\theta(s)} =$

$\frac{K(1+T_z s)}{s(1+T_{p1}s)(1+T_{p2}s)}$, como se la aprecia en la Figura 36a, mediante la relación del sistemas de

tercer orden con 2 polos, un cero y un integrador, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 1.01% como se observa en la Figura 36b.

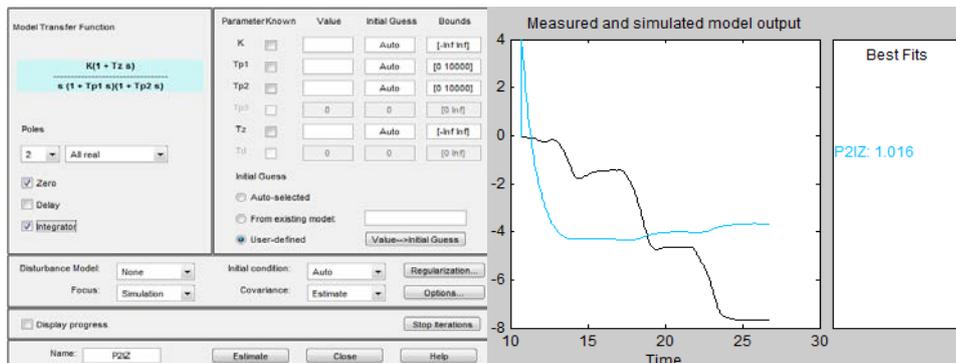


Figura 36a. Estructura del modelo 2 polos, 1 cero, 1 integrador posición `x`

Figura 36b. Estimación de tercer orden 2 polos, 1 cero, 1 integrador posición `x`

CAPÍTULO 3

Estimación añadiendo 3 polos, su estructura describe: $\frac{X(s)}{\theta(s)} = \frac{K}{(1+T_{p1}s)(1+T_{p2}s)(1+T_{p3}s)}$, como se la aprecia en la Figura 37a, mediante la relación del sistemas de tercer orden con 3 polos, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 43.22% como se observa en la Figura 37b.

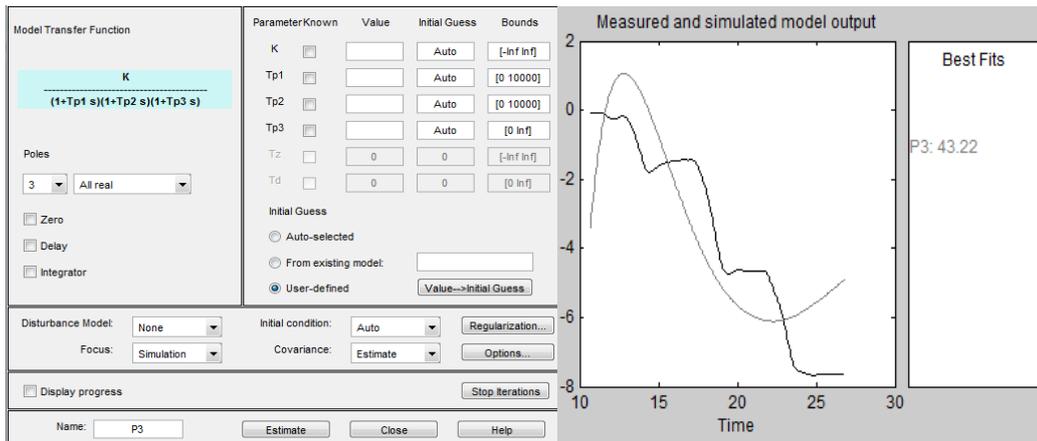


Figura 37a. Estructura del modelo 3 polos, posición `x`

Figura 37b. Estimación de tercer orden 3 polos, posición `x`

Estimación añadiendo 3 polos un cero, su estructura describe: $\frac{X(s)}{\theta(s)} = \frac{K(1+T_zs)}{(1+T_{p1}s)(1+T_{p2}s)(1+T_{p3}s)}$, como se la aprecia en la Figura 38a, mediante la relación del sistemas de tercer orden con 3 polos un cero, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 26.78% como se observa en la Figura 38b.

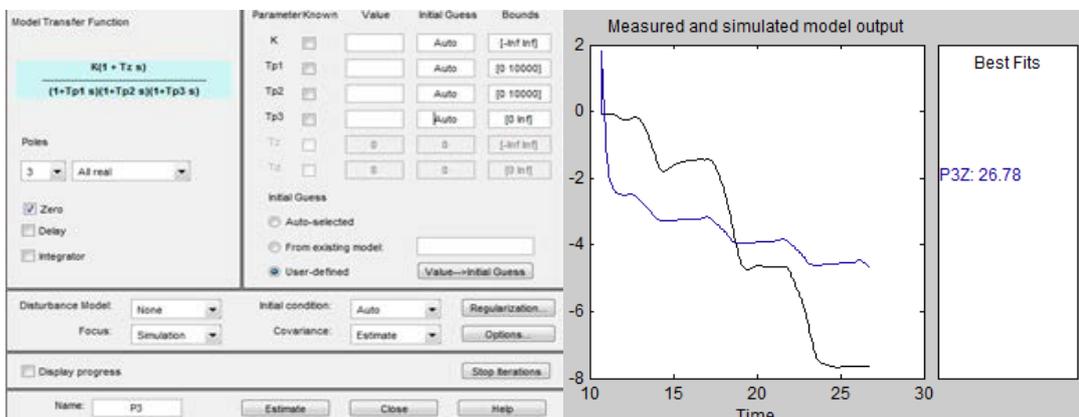


Figura 38a. Estructura del modelo 3 polos, 1 cero, posición `x`

Figura 38b. Estimación de tercer 3 polos, 1 cero, posición `x`

CAPÍTULO 3

Estimación añadiendo 3 polos un cero y un integrador, su estructura describe: $\frac{X(s)}{\theta(s)} =$

$$\frac{K(1+T_z s)}{s(1+T_{p1}s)(1+T_{p2}s)(1+T_{p3}s)}$$

, como se la aprecia en la Figura 39a, mediante la relación del sistemas de cuarto orden con 3 polos un cero y un integrador, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 59.04% como se observa en la Figura 39b.

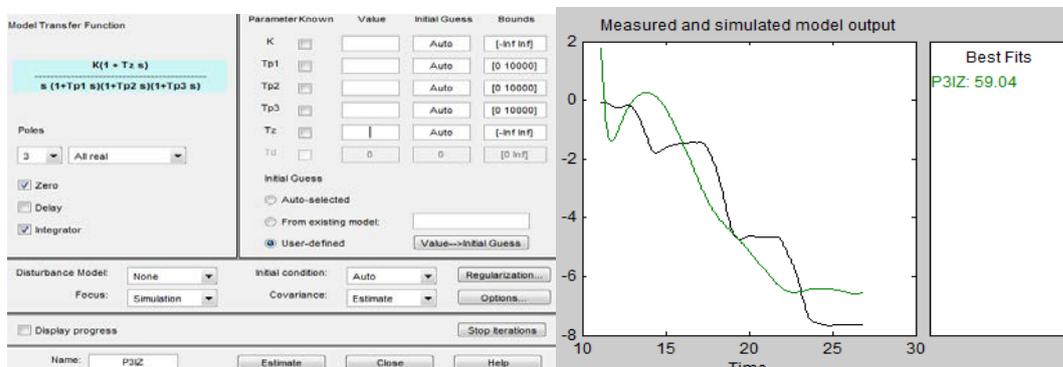


Figura 39a. Estructura del modelo 3 polos, 1 cero, 1 integrador, posición 'x' Figura 39b. Estimación de cuarto 3 polos, 1 cero, 1 integrador, posición 'x'

Sistema Discreto modelo ARX, Matlab permite generar modelos arx mediante la programación de las ecuaciones de estado del sistema usando sus librerías propias. No obstante y debido a que el tiempo de compilación es excesivo y se debe simular el modelo muchas veces antes de encontrar el modelo que más se ajusta a la realidad, se implementa el modelo arx de primer orden para la posición x.

Se elige un modelo arx111, su estructura describe: $\frac{X(z)}{\theta(z)} = \frac{b_1 z^{-1}}{1+a_1 z^{-1}}$, como se la aprecia en la

Figura 40a, la aproximación que presenta el modelo es de un 32,54%, como se observa en la Figura 40b.

CAPÍTULO 3

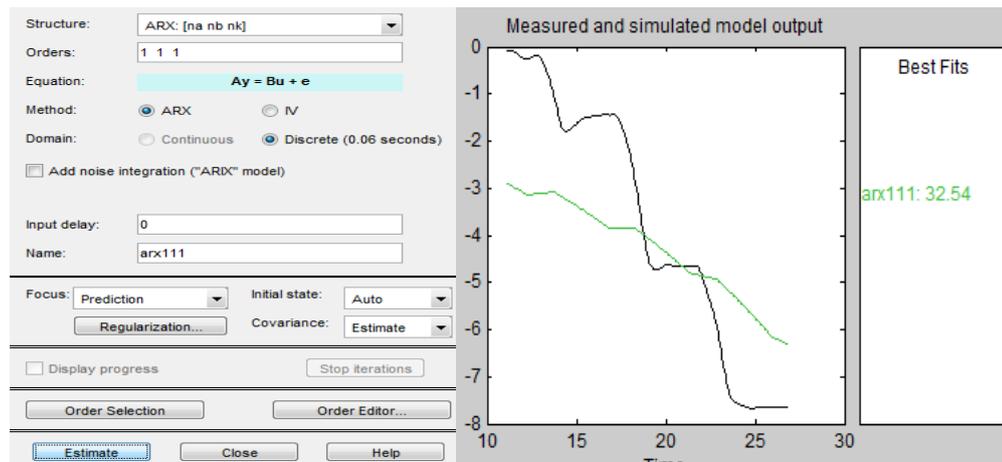


Figura 40a. Modelo ARX111, posición `x` Figura 40b. Modelo ARX 111, posición `x`

Se implementa el modelo arx121, su estructura describe: $\frac{x(z)}{\theta(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$,

como se la aprecia en la Figura 41a, la aproximación que presenta el modelo es de un -143,2%, como se observa en la Figura 41b.

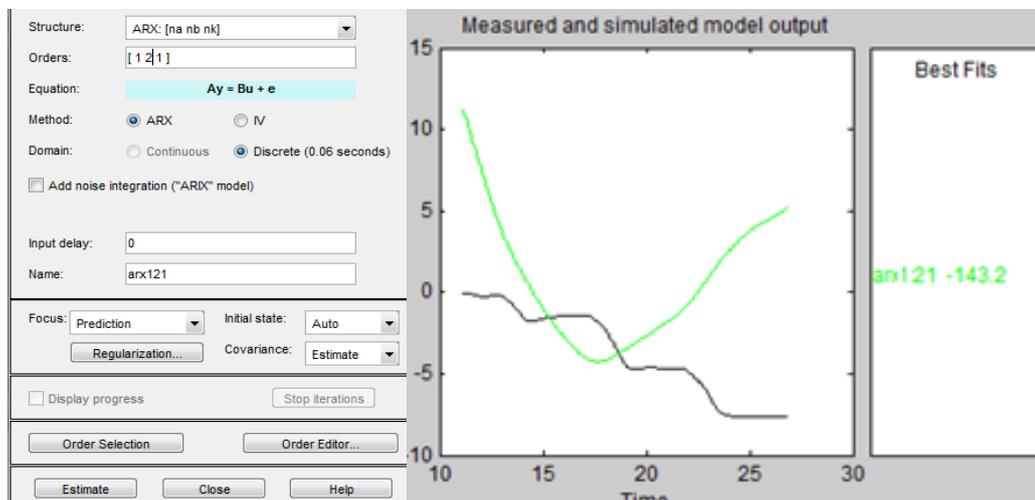


Figura 41a. Modelo ARX121, posición `x` Figura 41b. Modelo ARX 121, posición `x`

Con estos criterios se obtiene precisiones del 59.04% (Figura 39b), para el modelo en tiempo continuo, y precisiones de 32.54% para el modelo en tiempo discreto, se utiliza el modelo en tiempo continuo de cuarto orden, 3 polos un cero y un integrador, es el que más se aproxima al sistema real.

CAPÍTULO 3

El modelo identificado de la Figura 39b, correspondiente muestra la función de transferencia estimado como indica la Figura 42a.

```

Process model with transfer function:
          1+Tz*s
G(s) = Kp * -----
          s(1+Tp1*s)(1+Tp2*s)(1+Tp3*s)

      Kp = 3.3318
      Tp1 = 5.3816
      Tp2 = 0.36237
      Tp3 = 1.8607
      Tz = 4.5429

Name: P3IZ
Parameterization:
'P3IZ'

```

Figura 42a. Modelo de la ecuación general de cuarto orden estimado, posición `x`

Donde los parámetros que presenta la ecuación general son;

Kp: la constante de proporcionalidad

Tz: es el cero que arroja la identificación del modelo

Tp1: primero polo del sistema

Tp2: segundo polo del sistema

Tp3: tercer polo del sistema

S: integrador

Los conjuntos de valores o los modelos creados mediante el interfaz gráfico generalmente no están visibles desde el Workspace. Sin embargo, esta información puede ser exportada en cualquier momento al Workspace, ejecutando el comando `P3IZ = pem(G_identp,'P3IZ');` `Gc=tf(P3IZ)`, en la consola de Matlab se obtiene la ecuación del modelo estimado como se observa en la Figura 42b.

```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
>> tf(P3IZ)

ans =

    From input "u1" to output "y1":
          15.07 s + 3.318
    -----
    3.629 s^4 + 12.64 s^3 + 7.605 s^2 + s

Name: P3IZ
Continuous-time transfer function.
fx

```

Figura 42b. Función de transferencia identificada, posición `x` en tiempo continuo.

CAPÍTULO 3

Por lo tanto la función de transferencia de la figura 29, es la planta identificada de posición `x` con respecto al sistema del ángulo pitch:

$$P_x(s) = \frac{X(s)}{\theta(s)} = \frac{15.14s + 3.332}{3.629s^4 + 12.64s^3 + 7.605s^2 + s} \quad \text{Ec. 1}$$

Dónde:

$X(s)$: es la posición en el eje `x`

$\theta(s)$: es el ángulo de referencia pitch

3.4.2 Modelo para la posición `y`

En la Figura 44a muestra las variables de entrada y salida del sistema, se elimina la parte no lineal del sistema para identificar el sistema como un sistema lineal como se aprecia en la Figura 43b.

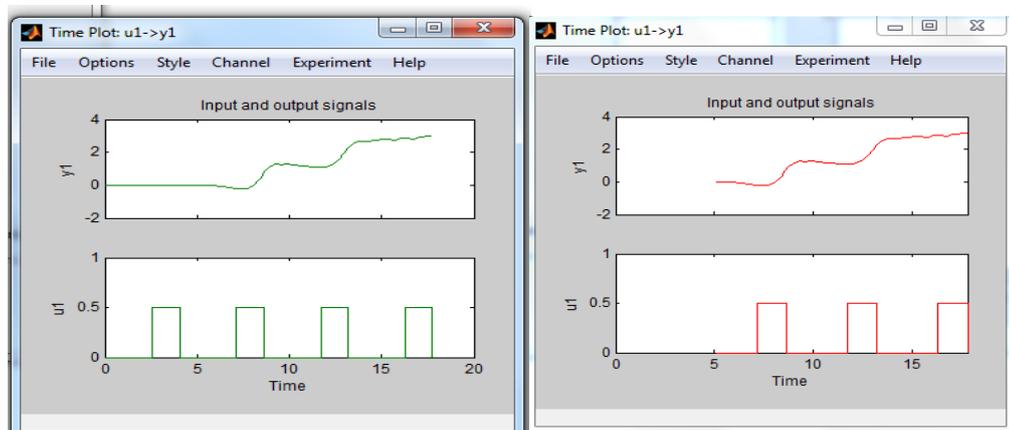


Figura 43a. Entrada y salida del sistema
Posición del eje `y`

Figura 43b. Eliminación tiempo muerto
Posición del eje `y`

Se elige la estructura añadiendo 1 polo, $\frac{y(s)}{\theta(s)} = \frac{K}{(1+T_{p1}s)}$, como se la aprecia en la Figura 44a, mediante la relación del sistema de primer orden con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -159.6% como se observa en la Figura 44b.

CAPÍTULO 3

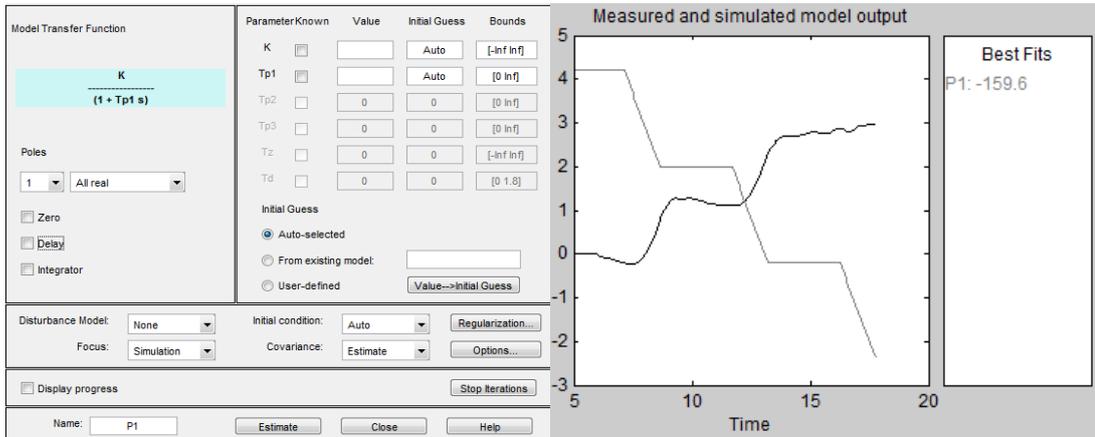


Figura 44a. Estructura del modelo acoplado 1 polo, posición 'y'

Figura 44b. Estimación Modelo de primer orden, posición 'y', calculado por Matlab

Estimación añadiendo 1 polo y un cero, su estructura describe: $\frac{y(s)}{\theta(s)} = \frac{K(1+T_zs)}{(1+T_{p1}s)}$, como se la aprecia en la Figura 45a, mediante la relación del sistemas de primer orden y un cero con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -192.3% como se observa en la Figura 45b.

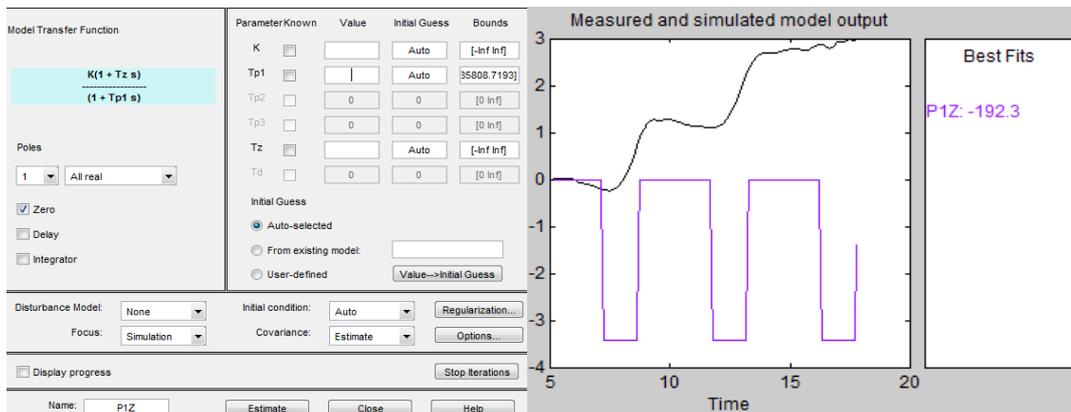


Figura 45a. Estructura del modelo acoplado 1 polo, 1 cero, posición 'y'

Figura 45b. Estimación Modelo de primer orden y 1 cero posición 'y'.

Estimación añadiendo 1 polo, 1 cero y un integrador, su estructura describe: $\frac{y(s)}{\theta(s)} = \frac{K(1+T_zs)}{s(1+T_{p1}s)}$, como se la aprecia en la Figura 46a, Mediante la relación del sistemas de segundo orden con

CAPÍTULO 3

un cero y un integrador con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -248.6% como se observa en la Figura 46b.

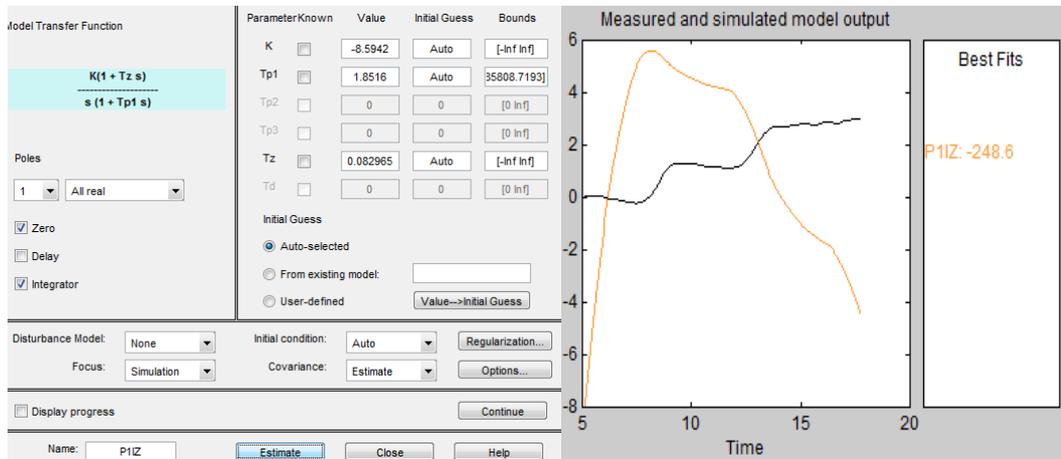


Figura 46a. Estructura del modelo 1 polo, 1 cero y 1 integrador posición `y`

Figura 46b. Estimación de segundo orden 1 polo, 1 cero y 1 integrador posición `y`

Estimación añadiendo 2 polos, su estructura describe: $\frac{y(s)}{\theta(s)} = \frac{K}{(1+T_{p1}s)(1+T_{p2}s)}$, como se la

aprecia en la Figura 47a, Mediante la relación del sistema de segundo orden con 2 polos, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -7.301% como se observa en la Figura 47b.



Figura 47a. Estructura del modelo 2 polos, posición `y`

Figura 47b. Estimación de segundo orden 2 polos, posición `y`

CAPÍTULO 3

Estimación añadiendo 2 polos un cero, su estructura describe: $\frac{y(s)}{\emptyset(s)} = \frac{K(1+T_z s)}{(1+T_{p1}s)(1+T_{p2}s)}$, como se

la aprecia en la Figura 48a, mediante la relación del sistemas de segundo orden con 2 polos y un cero, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -7.35% Figura 48a.

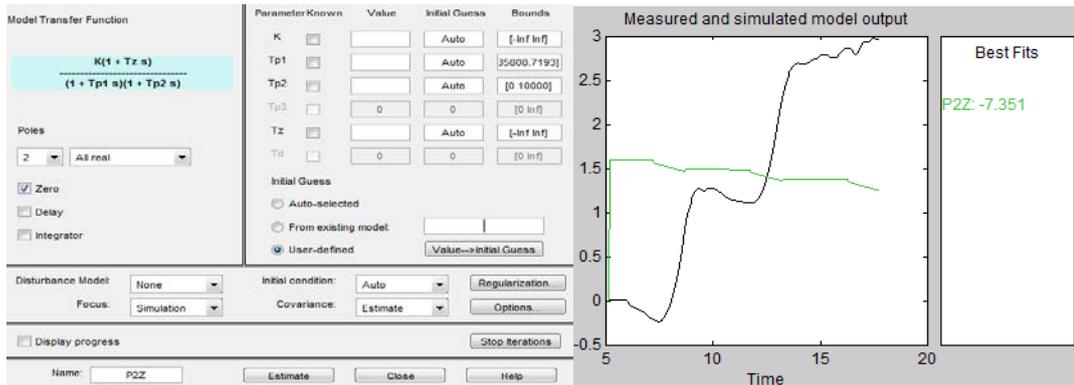


Figura 48a. Estructura del modelo 2 polos, 1 cero, posición `y`

Figura 48b. Estimación de segundo orden 2 polos, 1 cero, posición `y`

Estimación añadiendo 2 polos un cero y un integrador, su estructura describe: $\frac{y(s)}{\emptyset(s)} =$

$\frac{K(1+T_z s)}{s(1+T_{p1}s)(1+T_{p2}s)}$, como se la aprecia en la Figura 49a, mediante la relación del sistemas de

tercer orden con 2 polos, un cero y un integrador, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 83.73% como se observa en la Figura 49b.

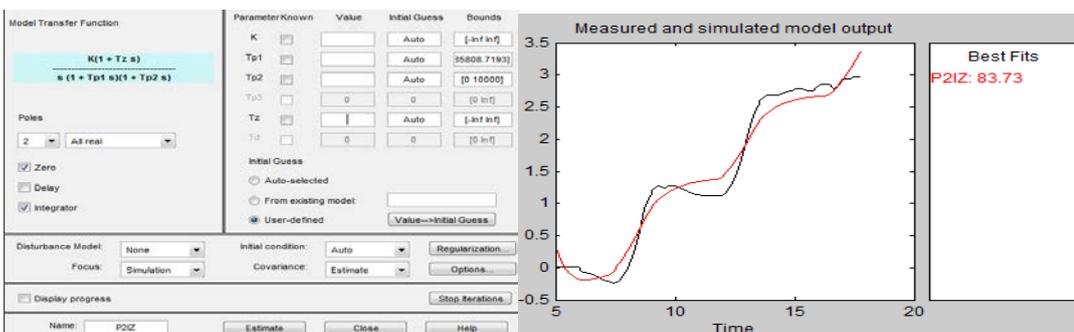


Figura 49a. Estructura del modelo 2 polos, 1 cero, 1 integrador posición `y`

Figura 49b. Estimación de tercer orden 2 polos, 1 cero, 1 integrador posición `y`

CAPÍTULO 3

Estimación añadiendo 3 polos, su estructura describe: $\frac{y(s)}{\theta(s)} = \frac{K}{(1+T_{p1}s)(1+T_{p2}s)(1+T_{p3}s)}$, como se la aprecia en la Figura 50a, mediante la relación del sistemas de tercer orden con 3 polos, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un -4.86e4% como se observa en la Figura 50b.

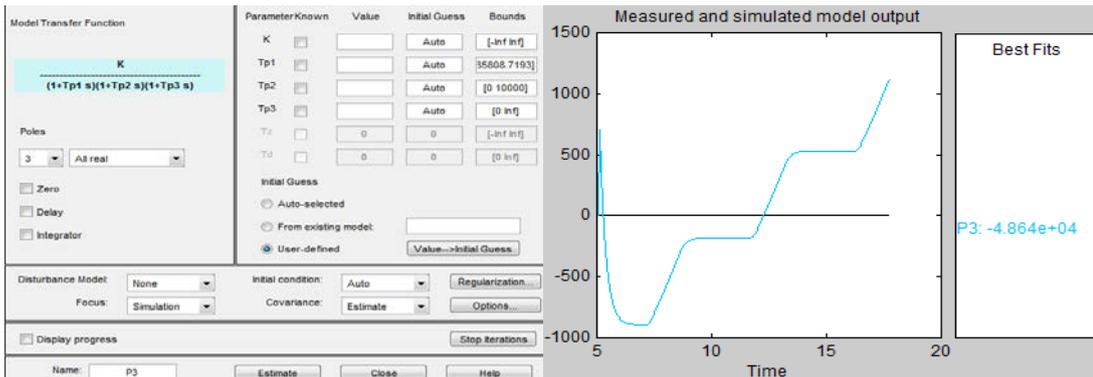


Figura 50a. Estructura del modelo 3 polos, posición `y`

Figura 50b. Estimación de tercer orden 3 polos, posición `y`

Estimación añadiendo 3 polos un cero, su estructura describe: $\frac{y(s)}{\theta(s)} = \frac{K(1+T_zs)}{(1+T_{p1}s)(1+T_{p2}s)(1+T_{p3}s)}$, como se la aprecia en la Figura 51a, mediante la relación del sistemas de tercer orden con 3 polos un cero, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 26.78% como se observa en la Figura 51b.

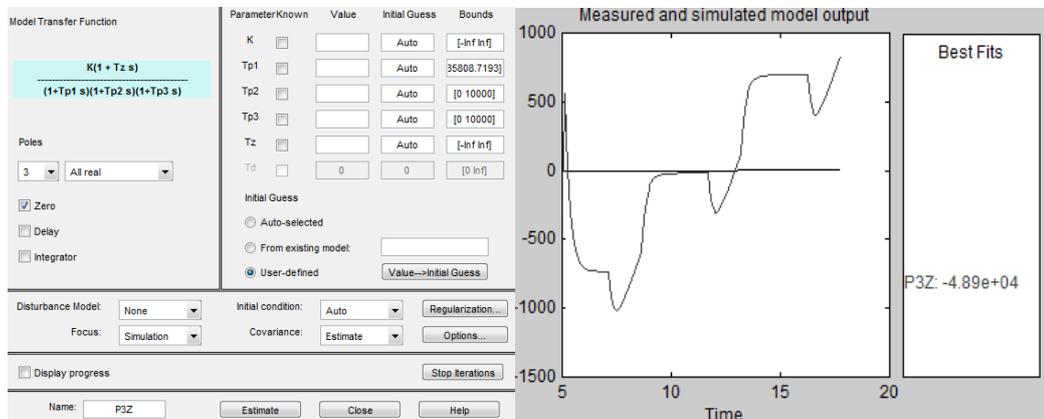


Figura 51a. Estructura del modelo 3 polos, 1 cero posición `y`

Figura 51b. Estimación de tercer orden 3 polos, 1 cero, posición `y`

CAPÍTULO 3

Estimación añadiendo 3 polos un cero y un integrador, su estructura describe: $\frac{y(s)}{\theta(s)} =$

$$\frac{K(1+T_z s)}{s(1+T_{p1}s)(1+T_{p2}s)(1+T_{p3}s)}$$

, como se la aprecia en la Figura 52a, mediante la relación del sistemas

de cuarto orden con 3 polos un cero y un integrador, con respecto a la planta, esta presenta una correlación con respecto a los valores experimentales en un 64.72% como se observa en la Figura 52b.

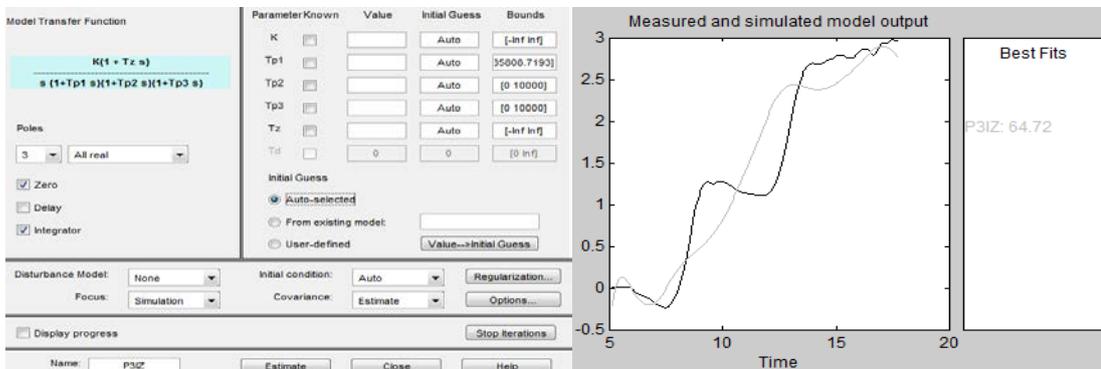


Figura 52a. Estructura del modelo 3 polos, 1 cero, 1 integrador posición 'y'

Figura 52b. Estimación de tercer orden 3 polos, 1 cero, 1 integrador posición 'y'

Modelo ARX;

Se elige un modelo arx111, su estructura describe: $\frac{y(z)}{\theta(z)} = \frac{b_1 z^{-1}}{1+a_1 z^{-1}}$, como se la aprecia en la

Figura 53a, la aproximación que presenta el modelo es de un -15,28%, como se observa en la Figura 53b.

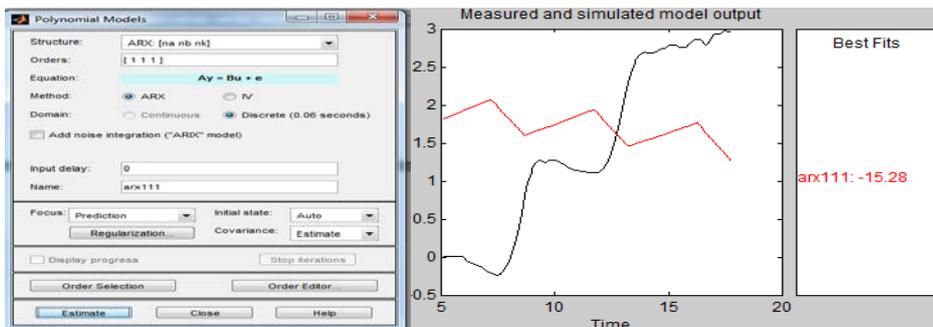


Figura 53a. Modelo ARX111 posición 'y'

Figura 53b. Modelo ARX 111, posición 'y'

CAPÍTULO 3

Se implementa el modelo arx121, su estructura describe: $\frac{y(z)}{\emptyset(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$.

La aproximación que presenta el modelo es de un -21,97%, como se observa en la Figura 54b.

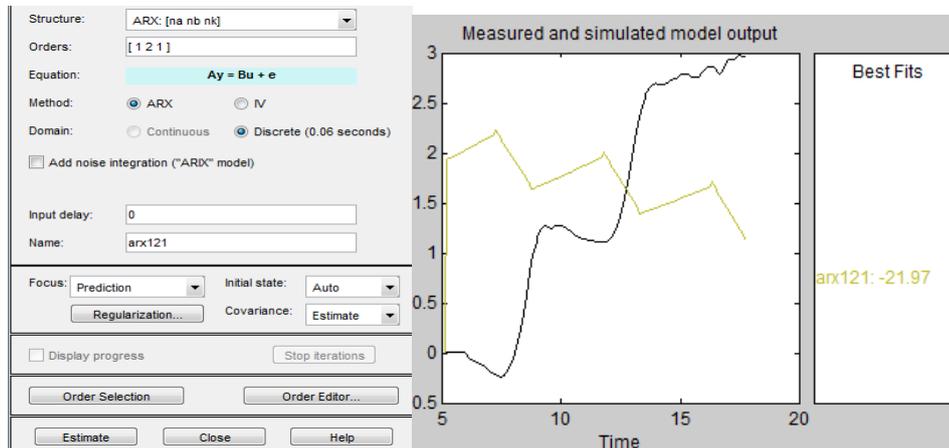


Figura 54a. Modelo ARX121 posición `y`

Figura 54b. Modelo ARX 121, posición `y`

Con estos criterios se obtiene precisiones del 83.73% (Figura 50b), para el modelo en tiempo continuo, y precisiones de -15.28% para el modelo en tiempo discreto, se utiliza el modelo en tiempo continuo de tercer orden, 2 polos un cero y un integrador, es el que más se aproxima al sistema real.

El modelo identificado de la Figura 50b, correspondiente muestra la función de transferencia estimado como indica la Figura 54c.

```

Process model with transfer function:
          1+Tz*s
G(s) = Kp * -----
          s(1+Tp1*s)(1+Tp2*s)

      Kp = 1.712
      Tp1 = 1.5021
      Tp2 = 0.7341
      Tz = 1.318
  
```

Figura 54c. Modelo de la ecuación general de cuarto orden estimado, posición `y`

Donde los parámetros que presenta la ecuación general son;

Kp: la constante de proporcionalidad

CAPÍTULO 3

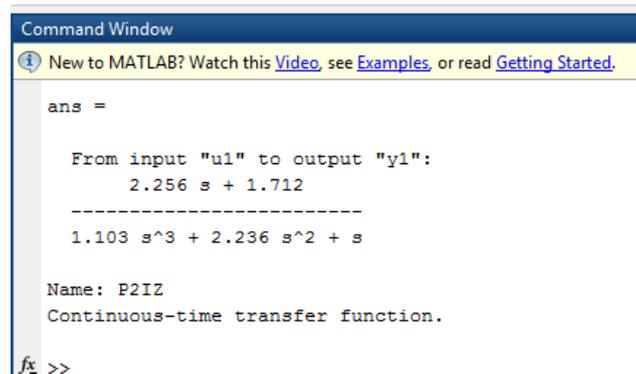
Tz: es el cero que arroja la identificación del modelo

Tp1: primero polo del sistema

Tp2: segundo polo del sistema

S: integrador

Los conjuntos de valores o los modelos creados mediante el interfaz gráfico generalmente no están visibles desde el Workspace. Sin embargo, esta información puede ser exportada en cualquier momento al Workspace, ejecutando el comando `P3IZ = pem(G_identp,'P3IZ');` `Gc=tf(P3IZ)`, en la consola de Matlab se obtiene la ecuación del modelo estimado como se observa en la Figura 54d.



```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

ans =

    From input "u1" to output "y1":
      2.256 s + 1.712
    -----
    1.103 s^3 + 2.236 s^2 + s

Name: P2IZ
Continuous-time transfer function.

fx >>

```

Figura 54d. Función de transferencia identificada, posición `y` en tiempo continuo.

Por tanto la función de transferencia de la figura 30, es la planta para la posición `y` con respecto al sistema del ángulo roll:

$$R_y(s) = \frac{y(s)}{\emptyset(s)} = \frac{2.256s + 1.712}{1.103s^3 + 2.236s^2 + s} \quad \text{Ec. 2}$$

Dónde:

$y(s)$: es la posición en el eje `y`

$\emptyset(s)$: es el ángulo de referencia roll

CAPÍTULO 4

DISEÑO DE CONTROL DE POSICIÓN DEL CUADRICÓPTERO

4.1 Sistema de control

El control de un cuadricóptero puede realizarse de muchas maneras, pero generalmente se realiza por medio de un ordenador, también conocido como unidad de control o controlador.

El controlador se encarga de generar acciones de control por medio de las variables manipuladas, a partir de los errores en las variables controladas medidas y sus referencias, que regulan la posición en x , y posición en y .

Un sistema de control es la combinación de componentes que actúan juntos para realizar el control de un proceso.

4.1.1.1 Objetivos del control

Se ensayan varios controles para conseguir una rápida respuesta, evitando todo lo que sea posible la sobre oscilación, y logrando una rápida estabilización.

Se exponen las principales técnicas de control usadas en cuadricópteros. Los cuadricópteros son inestables por sí mismos y se rigen por unas ecuaciones no lineales, de tal manera que es necesario diseñar un regulador de control para que la planta sea estable en lazo cerrado. En general, el objetivo de la teoría de control es el diseñar estrategias que permitan comandar un conjunto de variables manipuladas explicadas en el capítulo 2, de manera que se puedan mantener las variables controladas en unos valores deseados a pesar de las perturbaciones que puedan afectar al sistema.

4.1.1.2 Estrategias de control

La estrategia de control se lo hace como un sistema realimentado de simple entrada simple salida (SISO), y lineal, al igual que en el capítulo 3 sección 3.2, para la posición del eje x , y la posición del eje y , como se ilustra en las Figuras 55a y 55b.

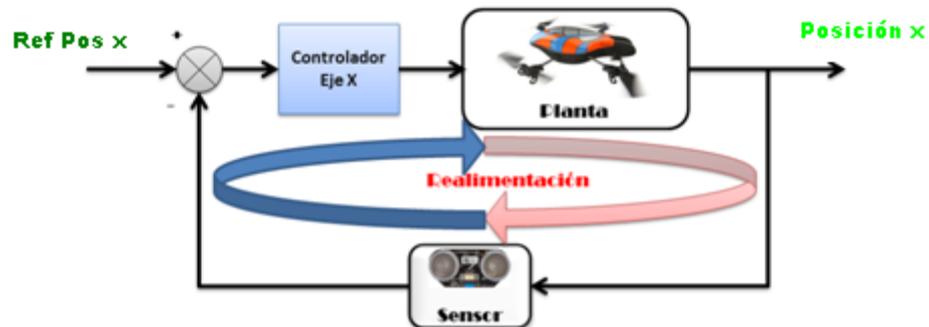


Figura 55a. Control en lazo cerrado posición x

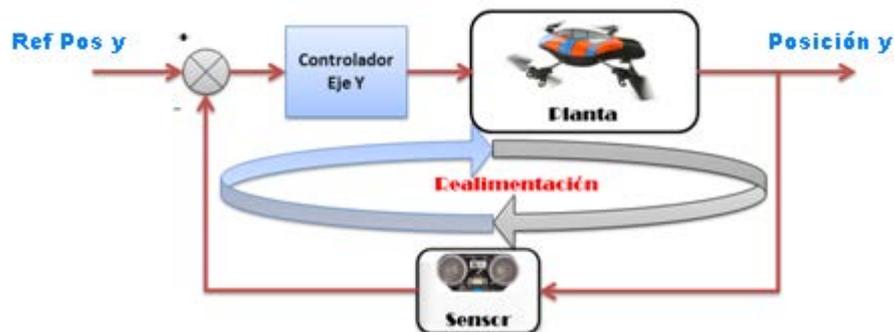


Figura 55b. Control en lazo cerrado posición y

Las estrategias de controles lineales y deterministas no son suficientes para controlar la dinámica compleja no lineal del cuadricóptero. Estas estrategias proponen diferentes algoritmos lineales según el punto de trabajo del sistema asociándose a cada régimen de vuelo (despegue, lateral, vertical), Como uno de los métodos de control lineal más usados se encuentran los controles PID.

4.1.1 Componentes del sistema

Se dispone de diversos componentes del cuadricóptero, a continuación se enlistan cada uno de ellos y el detalle de los mismos Figura 56.

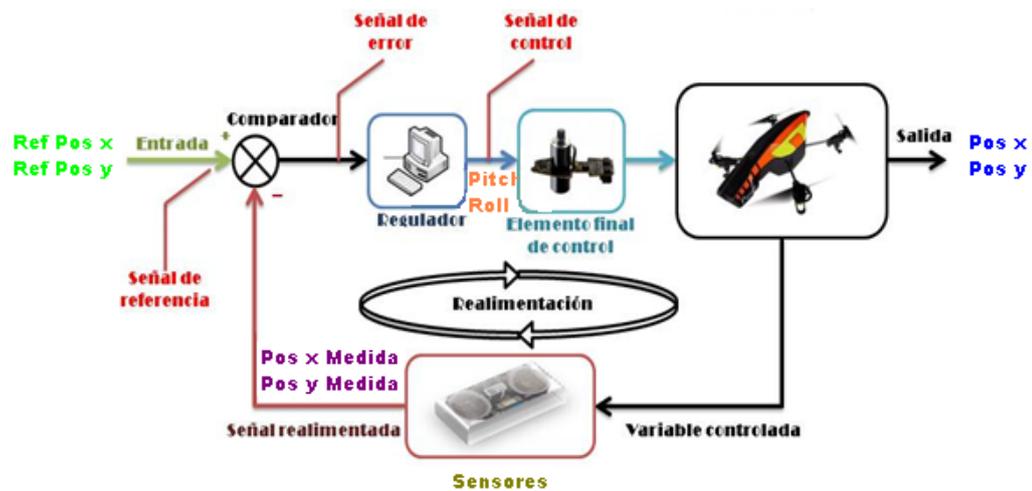


Figura 56. Componentes del cuadricóptero.

Sensores y transductores; estos dispositivos perciben el entorno geográfico en el que se mantiene el cuadricóptero, genera señales enviadas al espacio de vuelo con la intención de medir todas las variables físicas destinadas para la supervisión y control del vehículo. La señal de referencia pitch o roll se aplica a un dispositivo llamado detector de error, el captador compara su valor con el de la salida (posición medida), a través del lazo de realimentación, este componente mide en cada instante el valor de la magnitud de salida y provee una señal a dicho valor. El sensor de ultrasonido determina los valores iniciales midiendo distancias de altitud con respecto al suelo, el giroscopio permite medir el desplazamiento angular del cuadricóptero en el momento dado y al igual representa una medición estimada de la posición, el sensor acelerómetro detecta cambios de velocidad en el instante de acción de movimiento en el aire del vehículo.

Comparador; físicamente es un conjunto de dispositivos electrónicos que se encuentran alojados en la placa electrónica principal del AR.Drone 2.0, determina la señal de error que se entrega al regulador del sistema de control de posición del cuadricóptero, a través de la diferencia de la señal de salida deseada y la obtenida de forma experimental.

Regulador o controlador; es el cerebro del proyecto, la parte sustancial del sistema de control aplicado a la parte cinemática del AR.Drone 2.0. Este regulador establece una acción de control a través de la investigación matemática de sus parámetros en el instante de tiempo $t-1$, del comportamiento de estos parámetros y del modelado del sistema depende las variables de control destinadas hacia el regulador. El regulador se encuentra diseñado en el computador, y la algoritmia generada para el cálculo de sus variables se verá aplicada sobre la acción de movimiento del cuadricóptero. Estas variables son representadas por la acción proporcional, la acción integral y la acción derivativa, variables que mantienen constante la acción de control dentro de la planta del cuadricóptero.

Actuador; o también denominado elemento final de control, son los mecanismos que actúan sobre el proceso o planta según la señal de mando que reciba del regulador.

4.1.2 Controlador

Se puede utilizar casi cualquier tipo de controlador lineal como PI²⁷, PID²⁸, la señal de control debe ser una función lineal de los parámetros, suponiendo que el modelo es invariante con el tiempo también se necesita escoger un controlador adecuado.

Para controlar la posición del cuadricóptero se utiliza un controlador PID independiente para cada eje (x, y). Figuras 55a y 55b.

²⁷ PI Proporcional Integral

²⁸ PID Proporcional Integral Derivativo

CAPÍTULO 4

Un PID es un mecanismo de control por realimentación, corrige el error entre un valor medido y el valor que se quiere obtener, calculando y proporcionando una acción correctora que puede ajustar al proceso. El algoritmo de cálculo del control PID se da en tres acciones distintas: el proporcional, integral y derivativa, el valor Proporcional determina la reacción al error actual, el integral genera una corrección proporcional a la integral del error, se asegura que aplicando un esfuerzo de control suficiente el error de seguimiento se reduce a cero, el derivativo determina la reacción del tiempo en el que el error se produce.

La tabla 4 resume los efectos que tiene cada controlador en un sistema en lazo cerrado

Tabla 4: Efectos de las constantes PID en un sistema en lazo cerrado

	Tiempo de Subida	Sobre disparo	Tiempo de Establecimiento	Error de estado Estacionario
K_P	Disminuye	Aumenta	-	Disminuye
K_I	Disminuye	Aumenta	Aumenta	Aumenta
K_D	-	Disminuye	Disminuye	-

Fuente: (Ogata, 1998).

Elaborado por: Autores del proyecto

Las relaciones de la tabla 4 deben ser usadas sólo a modo de referencia, ya que las variables K_P , K_I y K_D son interdependientes, si se cambia el valor de una de estas variables, puede que cambie el efecto que producían las otras dos en el sistema, teniendo que reajustar su valor.

La suma de estas tres acciones es usada para ajustar al proceso, el controlador puede proveer un control diseñado para lo que requiera el proceso a realizar, en la Figura 57 muestra las ecuaciones que componen un control PID y la relación entre ellas:

CAPÍTULO 4

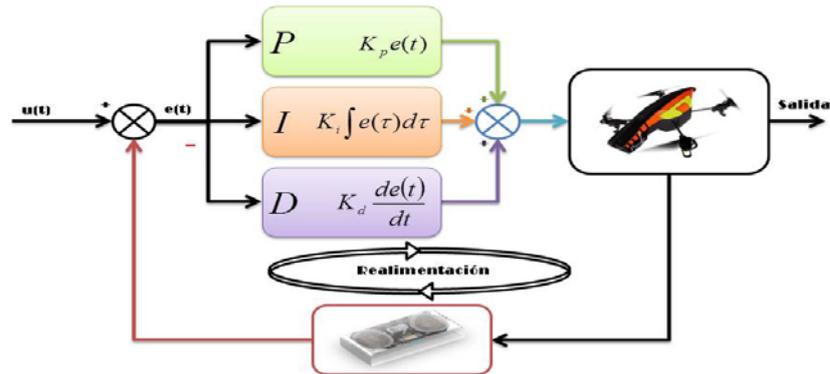


Figura 57. Configuración genérica de un controlador PID.

Algoritmo PID Continuo, se requiere el diagrama de lazo cerrado para iniciar el diseño del sistema de control de los ejes (x, y), como se muestra en la Figura 58.

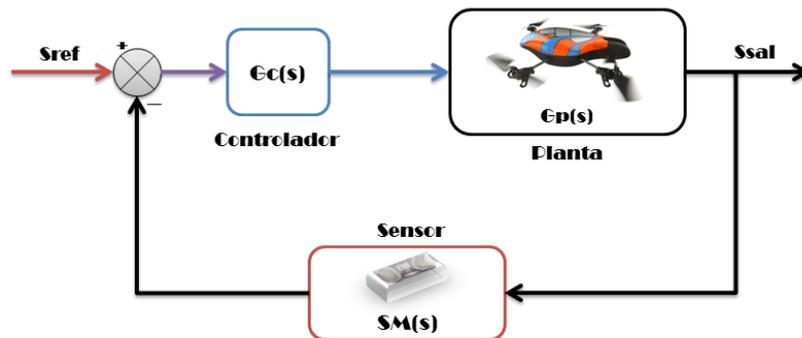


Figura 58. Diagrama del sistema de control en lazo cerrado.

Donde;

G_c : es el control PID

G_p : es la planta del sistema que representa a la máquina motriz o al excitador

SM : es el sensor de la señal de salida

S_{ref} : es la señal de entrada posición x, y deseados

S_{sal} : es la señal de salida posición x, y medidos

Como se puede observar en la Figura 58, el controlador está representado por $G_c(s)$, y la planta por $G_p(s)$ las cuales están en tiempo continuo.

Entonces el controlador PID se define por;

$$m(t) = k_p \left[e(t) + \frac{T_c}{T_i} \int_0^t e(t) dt + \frac{T_d}{T_c} * \frac{de(t)}{dt} \right] \quad \text{Ec. 3}$$

De donde se obtiene la función de transferencia;

$$G_c(s) = k_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad \text{Ec.4}$$

La ecuación 5 denota la función de transferencia del controlador PID en paralelo, para realizar la sintonización de los parámetros se utiliza la siguiente función de transferencia estándar, y posteriormente se obtiene los parámetros correspondientes.

$$G_c(s) = k_p \left(\frac{1 + T_i s + T_i T_d s^2}{s} \right) \quad \text{Ec.5}$$

Una vez obtenida la función de transferencia del controlador se requiere un método para el ajuste de los parámetros, se utiliza para los ejes (x, y), el método de sintonización de Ziegler-Nichols en lazo cerrado.

Este método se aplica cuando se tienen polos complejos conjugados, y consiste en;

- Añadir una constante proporcional K_c en el controlador del sistema como se observa en la Figura 58.
- Se eleva la ganancia, hasta que el sistema sea marginalmente estable para obtener la ganancia crítica K_{CR} .
- La ganancia crítica hace que el sistema oscile, por tanto se debe calcular el período de dicha oscilación, se lo conoce como periodo crítico T_{CR} , sabiendo que $s = j\omega$, $\omega = 2\pi f$ y $T = 1/f$.
- Se reemplazan los valores de K_{CR} y T_{CR} para hallar K_p , K_i y K_d , según la tabla 5, de Ziegler Nichols.

Tabla 5: Parámetros de ajuste del método de oscilación por Ziegles - Nichols

	Kp	Ti	Td
P	0.50 K _{CR}		
PI	0.45 K _{CR}	0.833 T _{CR}	
PID	0.60 K _{CR}	0.50 T _{CR}	0.125T _{CR}

Fuente: (Ogata, 1998).

Elaborado por: Autores del proyecto

PID método de sintonización de Ziegles - Nichols, este algoritmo PID se basa en la obtención de la ecuación en lazo cerrado para obtener el polinomio característico, para ello corresponde a:

$$G_{LC} = \frac{k_c G_p(s)}{1 + k_c G_p(s)} \quad \text{Ec. 6}$$

Por lo tanto el polinomio característico de la ecuación 12 es;

$$1 + k_c * G_p(s) = 0 \quad \text{Ec.7}$$

Arreglo de Routh - Hurwitz:

El criterio de estabilidad de Routh permite determinar la cantidad de polos en lazo cerrado que se encuentran en el semiplano derecho del plano s (raíces positivas) sin tener que factorizar el polinomio, este criterio de estabilidad sólo se aplica a los polinomios con una cantidad finita de términos.

Polinomio de s	Coeficientes del polinomio característico		
s ⁿ	a _n	a _{n-2}	a ₀ + K _c
s ⁿ⁻¹	a _{n-1}	a _{n-3}	
s ⁿ⁻²	b ₁	b ₂	
s ⁿ⁻³	c ₁		

CAPÍTULO 4

Elementos a calcular:

$$b_1 = \frac{a_{n-1} * a_{n-2} - a_n * a_{n-3}}{a_{n-1}} \quad \text{Ec.8}$$

$$b_2 = \frac{a_{n-1}(a_0 + K_c) - a_n * 0}{a_{n-1}} \quad \text{Ec.9}$$

$$c_1 = \frac{b_1 a_{n-3} - a_{n-1} b_2}{b_1} \quad \text{Ec.10}$$

Para obtener el K_{CR} ;

$$\frac{a_{n-1}(a_0 + K_c) - a_n * 0}{a_{n-1}} > 0 \quad \text{Ec.11}$$

Despejando k_c de la ecuación 12 se obtiene:

$$k_{CR} \geq -a_0 \quad \text{Ec. 12}$$

Para obtener el período crítico se reemplaza s por jw y el k_{CR} en el polinomio característico como indica la ecuación 19

$$a_n jw^n + a_{n-1} jw^{n-1} + a_{n-2} jw^{n-2} + a_{n-3} jw^{n-3} + k_{CR} = 0 \quad \text{Ec.13}$$

Las raíces del polinomio característico se obtiene por el comando de Matlab roots, obteniendo w que es la raíz del polinomio característico, se escoge la raíz positiva imaginaria; roots([a3 a2 a1 a0+k]), y se reemplaza la raíz en la ecuación 14.

$$T_{CR} = \frac{2\pi}{w} \quad \text{Ec.14}$$

Por último se reemplazan estos valores de K_{CR} y T_{CR} , para obtener los parámetros de la tabla 5, de esta manera se obtiene los parámetros del control PID.

Diseño del control PID para la posición `x`;

Se usa el método de sintonización de Ziegles-Nichols para el ajuste de los parámetros del controlador, determinando la ganancia crítica (K_{CR}), y el periodo crítico (T_{CR}), a partir de la ecuación 1 en lazo cerrado con realimentación unitaria, usando el criterio de Routh – Hurwitz.

Para el sistema en lazo cerrado con ganancia k.

$$\frac{K \left(\frac{15.14s + 3.332}{3.629s^4 + 12.64s^3 + 7.605s^2 + s} \right)}{1 + K \left(\frac{15.14s + 3.332}{3.629s^4 + 12.64s^3 + 7.605s^2 + s} \right)} = \frac{\frac{(15.14s + 3.332)K}{3.629s^4 + 12.64s^3 + 7.605s^2 + s}}{\frac{3.629s^4 + 12.64s^3 + 7.605s^2 + s + (15.14s + 3.332)K}{3.629s^4 + 12.64s^3 + 7.605s^2 + s}}$$

$$= \frac{(15.14s + 3.332)K}{3.629s^4 + 12.64s^3 + 7.605s^2 + (1 + 15.14K)s + 3.332K}$$

Ecuación característica en lazo cerrado:

$$3.629s^4 + 12.64s^3 + 7.605s^2 + (1 + 15.14K)s + 3.332K = 0 \quad \text{Ec. 15}$$

Desarrollando el sistema;

s^4	3.629	7.605	3.32k _c
s^3	12.64	1+15.14k _c	
s^2	$\frac{92.44 - 54.94k_c}{12.64}$	3.32k _c	
s^1	$\frac{-831.79k_c^2 + 814.16k_c + 92.44}{92.44 - 54.94k_c}$		
s^0	3.32k _c		

CAPÍTULO 4

Para la estabilidad, k_c debe ser positiva, y todos los coeficientes de la primera columna deben de serlo también.

$$k_c \geq 0$$

$$\frac{92.44 - 54.94k_c}{12.64} \geq 0$$

$$k_{CR} \leq 1.68$$

$$\frac{-831.79k_c^2 + 814.16k_c + 92.44}{92.44 - 54.94k_c} \geq 0$$

$$k_{CR1} \geq -0.102$$

$$k_{CR2} \geq 1.08$$

Para cualquier valor de k_c superior a 0 y menor a 1.68 el sistema será estable, y para cualquier valor mayor que 1.68 y menor a 0 será inestable, justo cuando $k_c = 0.13$ el sistema tendrá estabilidad marginal.

Por tanto, para que el sistema de control sea estable, el rango de k_c será:

$$0 \leq k_c \leq 1.68$$

Para $k_c = 0.13$ no hay ningún cambio de signo en los coeficientes en la primera columna, todos los coeficientes del polinomio característico son diferentes de cero y tienen el mismo signo, por tanto el sistema es asintóticamente estable.

Para obtener el periodo crítico se saca las raíces del polinomio ayudándonos del comando de Matlab; roots([3.629 12.64 7.605 (1 + 15.14 * k) 3.332 * k]), donde k=0.13, se obtiene:

$$w_1 = -0.2022 + j0.3692 \quad \text{Ec. 16}$$

$$w_2 = -0.2022 - j0.3692 \quad \text{Ec. 17}$$

$$w_3 = -2.8417 \quad \text{Ec. 18}$$

$$w_4 = -0.2370 \quad \text{Ec. 19}$$

CAPÍTULO 4

Se escoge la raíz positiva imaginaria

$$T_{CR} = \frac{2\pi}{0.3692} = 17.01 \quad \text{Ec. 20}$$

Se reemplaza la ganancia crítica (k_{CR}), y el periodo crítico (T_{CR}), para determinar los parámetros del controlador (tabla 5).

$$k_p = 0.6 * 0.13 = 0.078 \quad \text{Ec. 21}$$

$$T_i = 0.5 * 17.01 = 8.505 \quad \text{Ec. 22}$$

$$T_d = 0.125 * 17.01 = 2.126 \quad \text{Ec. 23}$$

La Figura 59 muestra el diagrama del sistema de control para la posición del eje `x`.

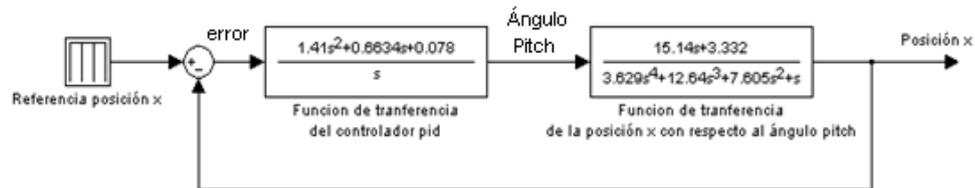


Figura 59. Control PID para el posicionamiento del eje `x`.

Diseño del control PID para la posición `y`;

A partir de la ecuación 2, en lazo cerrado con realimentación unitaria se halla la ganancia crítica (K_{CR}), y el periodo crítico (T_{CR}), para determinar los parámetros del controlador.

Para el sistema en lazo cerrado con ganancia k.

$$\begin{aligned} \frac{K \left(\frac{2.256s + 1.712}{1.103s^3 + 2.236s^2 + s} \right)}{1 + K \left(\frac{2.256s + 1.712}{1.103s^3 + 2.236s^2 + s} \right)} &= \frac{\frac{(2.256s + 1.712)K}{1.103s^3 + 2.236s^2 + s}}{\frac{1.103s^3 + 2.236s^2 + s + (2.256s + 1.712)K}{1.103s^3 + 2.236s^2 + s}} \\ &= \frac{(2.256s + 1.712)K}{1.103s^3 + 2.236s^2 + (1 + 2.256K)s + 1.712K} \end{aligned}$$

CAPÍTULO 4

Ecuación característica en lazo cerrado:

$$1.103s^3 + 2.236s^2 + (1 + 2.256K)s + 1.712K = 0 \quad \text{Ec. 24}$$

Desarrollando el sistema;

s^3	1.103	$1+2.256k_c$
s^2	2.236	$1.712k_c$
s^1	$\frac{3.16k_c+2.236}{2.236}$	
s^0	$1.712k_c$	

Cálculo de la ganancia crítica K_{CR} :

$$k_c \geq 0$$

$$\frac{2.236 + 3.16k_c}{2.236} \geq 0$$

$$k_{CR} \geq -0.71$$

Desarrollando las inecuaciones del sistema se obtiene que para cualquier valor de k_C mayor a 0 el sistema será estable, y para cualquier valor menor a 0 será inestable, justo cuando $k_C = 0.3$ el sistema tendrá estabilidad marginal.

Por tanto, para que el sistema de control sea estable, el rango de k_C será:

$$k_c \geq 0$$

Para $k_C = 0.3$ se determinó que todos los coeficientes de la primera columna no tiene ningún cambio de signo, los coeficientes del polinomio característico son diferente de cero y tienen el

CAPÍTULO 4

mismo signo, por las características del sistema se ajusta el controlador usando el método de Ziegles – Nichols en tiempo continuo.

Para obtener el periodo crítico se saca las raíces del polinomio ayudándonos del comando de Matlab; $\text{roots}([1.103 \ 2.236 \ (1 + 2.256 * k) \ 1.712 * k])$, donde $k=0.3$, se obtiene:

$$w_1 = -0.5418 + j0.447 \quad \text{Ec. 25}$$

$$w_2 = -0.5418 - j0.447 \quad \text{Ec. 26}$$

$$w_3 = -0.9316 \quad \text{Ec. 27}$$

Se escoge la raíz positiva imaginaria

$$T_{CR} = \frac{2\pi}{0.447} = 14.052 \quad \text{Ec. 28}$$

Se reemplaza la ganancia critica (k_{CR}), calculado y el periodo critico (T_{CR}), para determinar los parámetros del controlador (tabla 5).

$$k_p = 0.6 * 0.3 = 0.18 \quad \text{Ec. 29}$$

$$T_i = 0.5 * 14.052 = 7.026 \quad \text{Ec. 30}$$

$$T_d = 0.125 * 14.052 = 1.76 \quad \text{Ec. 31}$$

La Figura 60 muestra el diagrama del sistema de control para la posición del eje `y`.

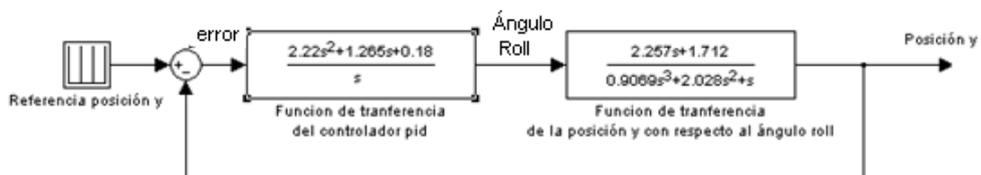


Figura 60. Control PID para el posicionamiento del eje `y`.

4.1.3 Simulación del sistema de control de posición en Matlab

La tarea del control de posición es poder medir la fiabilidad de los movimientos al enviarle una consigna de posición en el plano (x, y), ya que el algoritmo de control de posición en Matlab ayuda a visualizar en el computador que tan óptimo es, pero lo que no se sabe que tan preciso es en la realidad, es decir que si se simula una consigna de 1m en el eje `x`, y 1m en el eje `y`, se debe saber que en la realidad el cuadricóptero se moverá esa distancia, de lo contrario se debe modificar el algoritmo de control de posición en Matlab.

Para la simulación del control de posición en Matlab, el sistema de orientación está compuesta por las siguientes posiciones de referencia, y se realiza el análisis de las simulaciones por cada uno de los ejes (x, y), comparando tiempos de respuesta.

Posicionamiento para el eje `x`.

Se pondrá a prueba el posicionamiento del eje `x`, para esto se solicita al simulador de Matlab una consigna de 2, -2.

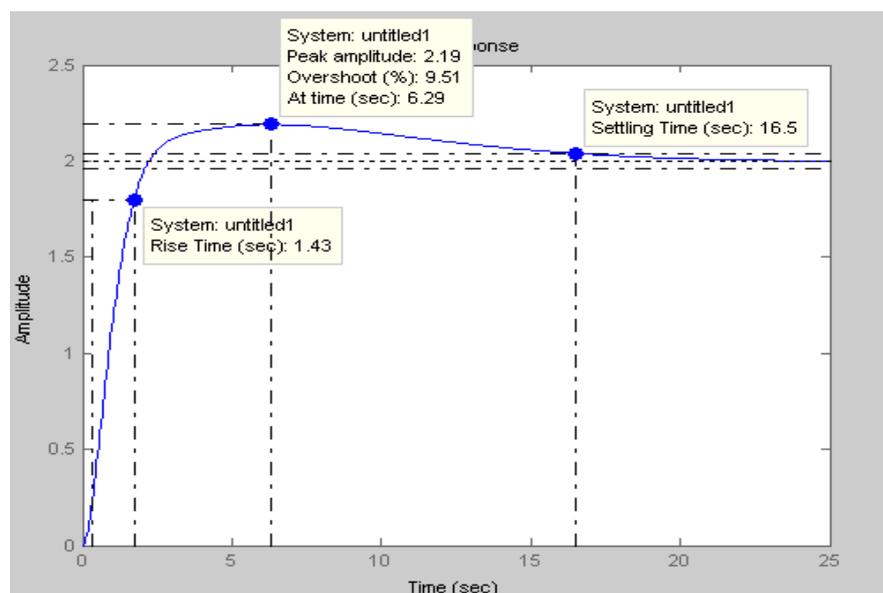


Figura 61. Respuesta simulada con una consigna de 2 para el eje `x`

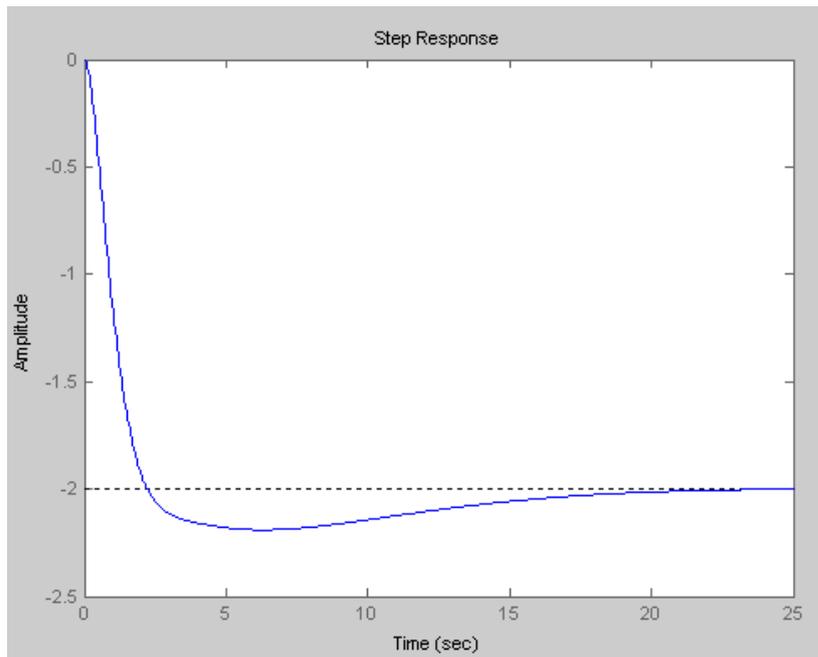


Figura 62. Respuesta simulada con una consigna de -2 para el eje `x`

Posicionamiento para el eje `y`.

Se pondrá a prueba el posicionamiento del eje `y`, para esto se solicita al simulador una consigna de 2, -2.

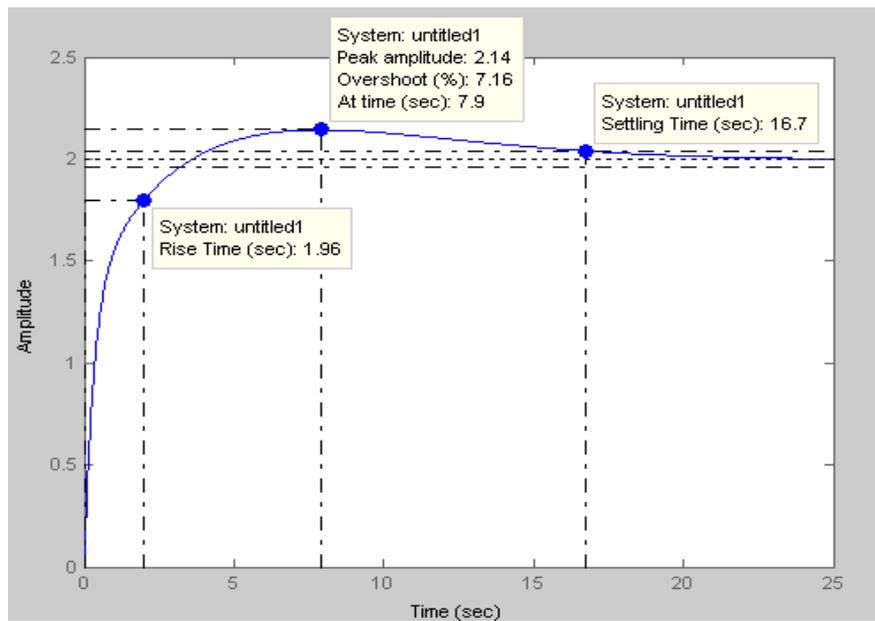


Figura 63. Respuesta simulada con una consigna de 2 para el eje `y`

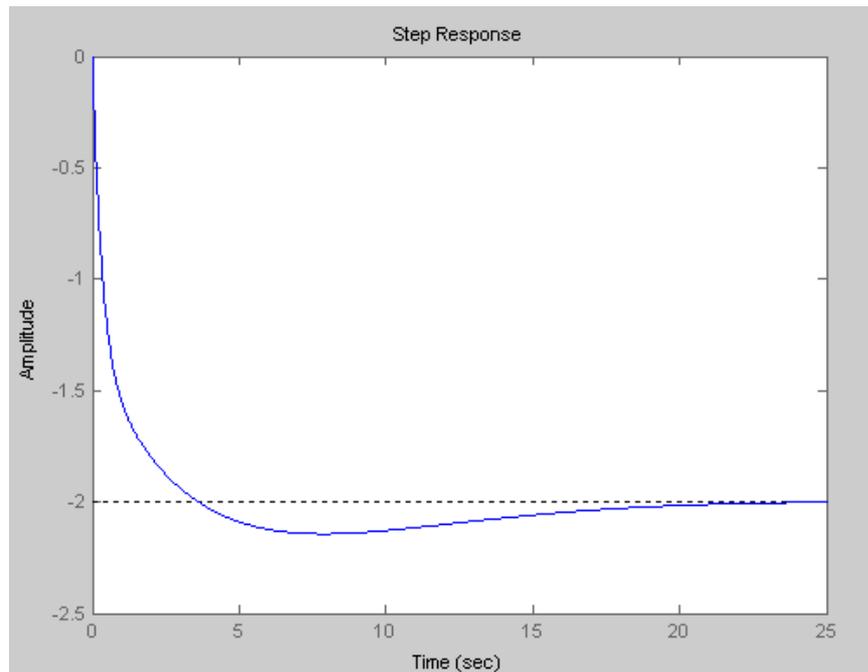


Figura 64. Respuesta simulada con una consigna de -2 para el eje `y`

Se ha diseñado estas pruebas para poder medir la fiabilidad de los movimientos del AR.Drone 2.0 al enviarle una consigna de posición en el plano XY.

Mediante la evaluación de los datos de respuesta obtenida se tiene el siguiente error en estado estable como se indica en la tabla 6.

Tabla 6: Respuesta en el tiempo para las posiciones de los ejes (x, y).

Posición	Distancia [m]	Tiempo de Establecimiento [s]	Sobre pico [%]
Eje x	2	16.5	9.51
Eje x	-2	16.5	9.51
Eje y	2	16.7	7.16
Eje y	-2	16.7	7.16

Elaborado por: Autores del proyecto

CAPÍTULO 4

Los controladores para las posiciones de los ejes (x, y), tienen un tiempo de establecimiento y sobre picos altos, por lo que se procede a ajustar los controladores para minimizar el error.

4.1.4 Modificaciones del sistema de control

La estrategia para mejorar las respuestas de los controladores es autoajustar los parámetros del control de posición de los ejes (x, y), manualmente hasta que se estabilice el sistema y disminuir el tiempo de establecimiento.

Ajuste de parámetros para el eje `x`.

Para obtener los parámetros ajustados se determina los efectos de las constantes PID y la relación de la tabla 4, para disminuir el tiempo de establecimientos se toma un polo en $w = 0,2$; teniendo como resultado un período crítico de $T_{CR} = 31.4$.

$$k_p = 0.6 * 0.13 = 0.078 \quad \text{Ec. 32}$$

$$T_i = 0.5 * 31.4 = 15.7 \quad \text{Ec. 33}$$

$$T_d = 0.125 * 31.4 = 3.93 \quad \text{Ec. 34}$$

La Figura 65 muestra el diagrama del sistema de control, con los parámetros ajustados para la posición del eje `x`.

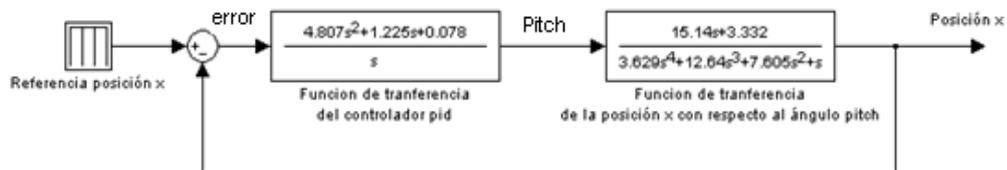


Figura 65. Control PID con parámetros ajustados posición eje `x`.

En este apartado se muestra la salida del sistema para las diferentes combinaciones de mejoras al diseño de control para el posicionamiento en el eje x. para lo cual se pone a prueba el diseño de la Figura 65, para esto se solicita al simulador una consigna de 2.

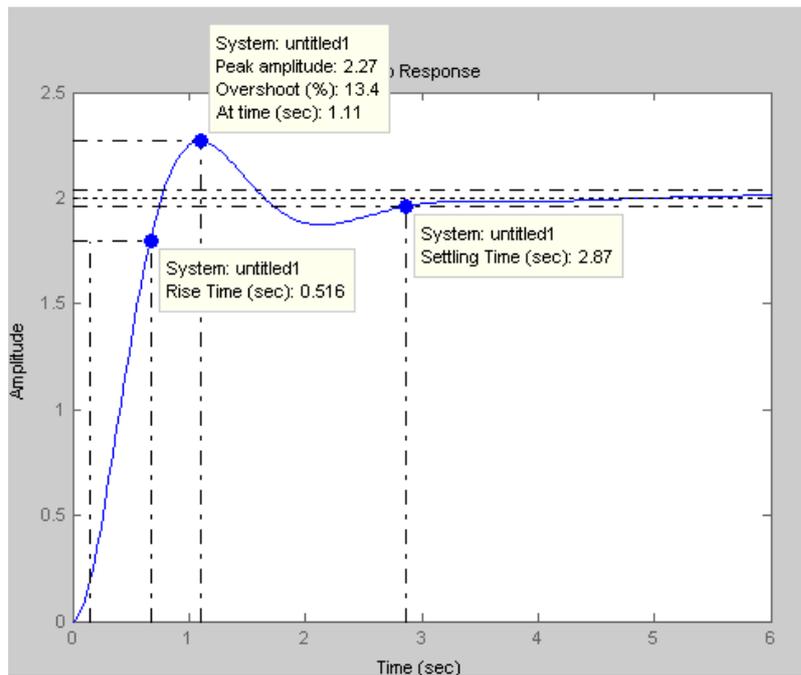


Figura 66. Parámetros autoajustados con una consigna de 2 para el eje `x`

Ajuste de parámetros para el eje `y`.

Para disminuir el tiempo de establecimientos se toma un polo en $w = 0,1$; teniendo como resultado un periodo crítico de $T_{CR} = 62.8$.

$$k_p = 0.6 * 0.3 = 0.18 \tag{Ec. 35}$$

$$T_i = 0.5 * 62.8 = 31.4 \tag{Ec. 36}$$

$$T_d = 0.125 * 62.8 = 7.85 \tag{Ec. 37}$$

La Figura 67 muestra el diagrama del sistema de control, con los parámetros ajustados para la posición del eje `y`.

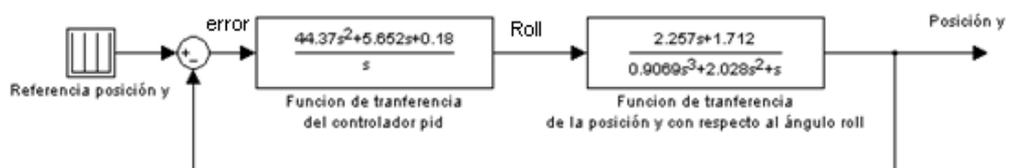


Figura 67. Control PID con parámetros ajustados posición eje `y`.

CAPÍTULO 4

Se pone a prueba el diseño de la Figura 67, para esto se solicita al simulador una consigna de 2.

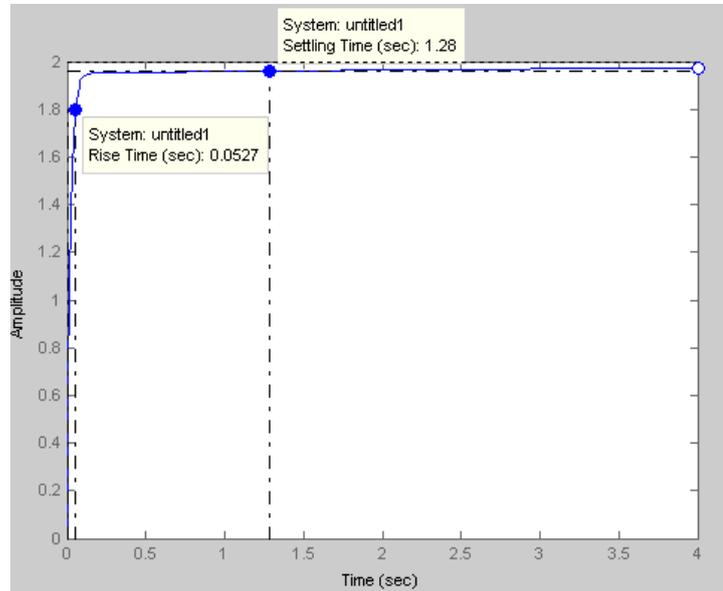


Figura 68. Parámetros autoajustados con una consigna de 2 para el eje `y`

Para evaluar las mejoras del diseño del controlador de posición de los ejes (x, y), la tabla 7 y 8, indica las comparaciones de error en estado estable.

Tabla 7: Comparación de respuesta en el tiempo posición del eje `x`.

Sistema	Distancia [m]	Tiempo de Establecimiento [s]	Sobre pico [%]
Sin mejora	2	16.5	9.51
Mejorado	2	2.87	13.4

Elaborado por: Autores del proyecto

Tabla 8: Comparación de respuesta en el tiempo posición del eje `y`.

Sistema	Distancia [m]	Tiempo de Establecimiento [s]	Sobre pico [%]
Sin mejora	2	16.7	7.16
Mejorado	2	1.28	0

Elaborado por: Autores del proyecto

CAPÍTULO 5

IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DEL CUADRICÓPTERO

5.1 Estudio y análisis de la interfaz humano/máquina

Una vez realizado el modelado y posterior diseño de control del cuadricóptero, la implementación de la parte matemática del modelamiento y control en el computador, se presenta a través de una herramienta importante de programación como es el programa TrackDrone Lite 2.0. Gracias a los desarrolladores, este programa beneficia en parte a este proyecto debido a su compatibilidad con el AR.Drone 2.0 de la familia Parrot, y dada sus prestaciones en la interfaz se logra visualizar la trayectoria, características de navegación, la maniobrabilidad y posterior control del cuadricóptero. De igual manera permite acondicionar parámetros importantes necesarios para el control de posición.

Desarrollado por el Grupo de Control Predictivo y Optimización Heurística, del Instituto Universitario de Automática e Informática Industrial UPV²⁹, el programa TrackDrone Lite 2.0 como se aprecia en la Figura 69, es una herramienta de investigación que permite la utilización de su algoritmo de programación, con la finalidad de proveer analogías de aplicación para proyectos de control en cuadricópteros (UPV, 2012).

²⁹ UPV Universidad Politécnica de Valencia.

CAPÍTULO 5



Figura 69. Entorno TrackDrone Lite 2.0.

La plataforma de programación está desarrollada en Labview, utilizando las herramientas primordiales de comunicación, el programa presenta una comunicación eficiente entre computador y cuadricóptero, esto es muy importante ya que en el instante de despegue del AR.Drone 2.0, el sistema entrega datos de navegación en tiempo real, el cual presenta datos de posición y velocidad en vuelo, desplazamientos en los ejes (x, y), de igual manera presenta una trayectoria estimada y/o acorde al desplazamiento real que presenta el cuadricóptero.

Este programa es ejecutable en los sistemas operativos Windows Xp/Windows 7, en algunos casos, debido a que los permisos de seguridad de administrador son altos en Windows 7, las ejecuciones de las entradas que se aplican al cuadricóptero no se ejecutan de manera normal y en algunas ocasiones es necesario guardar los archivos ejecutables en otra carpeta distinta a donde se encuentra la carpeta de instalación en el sistema, de igual manera se tiene que desactivar programas tales como antivirus, puertos de entradas, firewall, ya que el objetivo es que no exista interrupciones de comunicación con el cuadricóptero.

TrackDrone Lite 2.0 es un programa versátil en su utilización y programación, una herramienta eficaz para el enlace de comunicación entre el computador y el vuelo del

CAPÍTULO 5

cuadricóptero, manejo y algoritmos de control. El programa presenta al usuario 4 áreas importantes desarrolladas para presentar de manera ágil las ejecuciones en tiempo real del sistema total del cuadricóptero. Estas áreas de visualización comprende la parte de comunicación, la parte de desplazamiento y rutas de navegación, la parte de inicialización de vuelo del cuadricóptero, y la parte de manejo de aplicación de control y trayectorias, descritas más a fondo en la Figura 70.



Figura 70. Áreas de aplicación del programa.

Área de comunicación, presenta al usuario la visualización de los estados de inicialización. El primer cuadro de mensaje denominado “ARDrone IP” determina la comunicación del computador con el cuadricóptero, se lo hace a través de señal WIFI y de esta manera el cuadricóptero presenta una dirección IP que ayuda al enlace con el computador, por defecto todo firmware cargado en el sistema del AR.Drone 2.0 presenta una dirección IP 192.168.1.1 que será constante en todos los vehículos de la familia Parrot, se puede apreciar en la Figura 71.



Figura 71. Dirección IP representado en TrackDrone Lite 2.0

El segundo cuadro de mensaje determinado como “Acquisition Period” representa el tiempo de muestreo, se lo define en los 60mseg, con la finalidad de que el procesamiento de cálculo de ejecución de la aplicación de control sea eficiente llegando a un nivel cuasi óptimo.

Una parte importante en la ejecución de enlace computador/cuadrícóptero, es la visualización de los comando AT que ejecuta en tiempo real el cuadrícóptero. Para que exista un adecuado nivel de ejecución, existe un indicador “On Time Control Loop”, este indicador tiene la tarea de presentar al usuario si hay o no un buen lazo de control mediante el cambio de color, ya sea el color verde para un óptimo procesamiento y el color rojo indica un incorrecto procesamiento, ver Figura 72.



Figura 72. Tiempo de muestreo e indicador óptimo/fallo de lazo de control

Si por algún instante existe una pérdida de enlace de comunicación entre el cuadrícóptero y el computador, el TrackDrone Lite 2.0 propone al usuario dos botones destinados para el reinicio, el primer botón “Rest Comm”, utiliza un reinicio de las comunicaciones para establecer sincronía. En el encendido del cuadrícóptero su programación interna establece puntos de referencia y parámetros de navegación esto mediante sensores acoplados en el mismo, cuando existe una interferencia ya sea por movimientos bruscos, suelo irregular el

CAPÍTULO 5

AR.Drone 2.0 no podrá despegar indicando luces rojas como falla del sistema, para reparar al cuadricóptero el segundo botón denominado “Resume Drone”, se configura a las condiciones iniciales del AR.Drone 2.0, de esta manera ya no será necesario desconectar la batería y de una forma automática y más eficiente mediante el TrackDrone Lite 2.0 se podrá reiniciar su puntos de referencia antes del despegue, ver Figura 73.



Figura 73. Botones indicativos para la configuración de condiciones iniciales.

Área de inicialización de vuelo, esta área presenta botones destinados para la acción de arranque del sistema y la acción de navegación del cuadricóptero. El botón denominado “Start Flying” permite al computador iniciar el arranque del sistema del cuadricóptero verificando su estado inicial al usuario, y posterior realizar la acción de vuelo, muestra un indicador que presenta el mensaje “Disconnected/Connected” entendiéndose como una conexión de encendido del cuadricóptero. Los botones destinados para el despegue y aterrizaje del cuadricóptero son señalados como “Take Off” y “Land” respectivamente como se aprecia en la Figura 74. El botón “Exit” tiene la función principal de reiniciar el sistema completo del TrackDrone Lite 2.0 en casos donde existan problemas de enlace con el cuadricóptero, al accionar este botón automáticamente reinicia el sistema empezando desde el principio para una nueva utilización de programa.



Figura 74. Botones destinados para la acción de vuelo del cuadricóptero

El TrackDrone Lite 2.0 exhibe un botón definido como “UAV Mode/Manual” como se aprecia en la Figura 75a, y que permite maniobrar al AR.Drone 2.0 en dos modos de vuelo. Un modo de vuelo es manual, mediante este modo el control de vuelo del cuadricóptero se lo realiza a través del teclado del computador y se desarrollara el vuelo del cuadricóptero de forma manual, al mismo tiempo se tiene acceso a la ejecución de los mandos de aceleración como se aprecia en la Figura 75b, el cual da como resultado las inclinaciones de los ángulos de navegación Roll y Pitch, y un interruptor “Unidirection/Multidirection” el mismo que es destinado para el tipo de movimiento, al aplicar modo “Unidirection” el desplazamiento será en un simple eje, mientras que el modo “Multidirection” será un desplazamiento uniforme en todo el campo de vuelo.

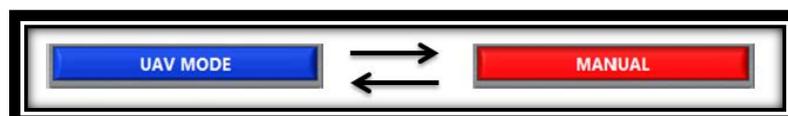


Figura 75a. Botón de elección modo manual y modo automático

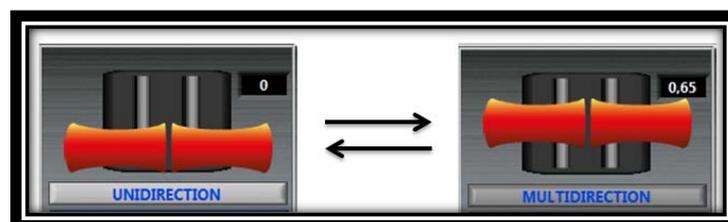


Figura 75b. Mando de aceleración para modo manual

El siguiente modo de vuelo es automático y es una herramienta importante, conlleva a adaptar aplicaciones de control en el vuelo del cuadricóptero, existe dos aplicaciones de control

CAPÍTULO 5

accionando el modo UAV. La acción de control en lazo abierto, esta aplicación presenta un control de movimientos en Pitch, Roll y también combinados, es decir se programa el movimiento del cuadricóptero para visualizar la trayectoria realizada por el mismo, es necesario pulsar el botón “Open Loop Control” el cual se aprecia en la Figura 76.

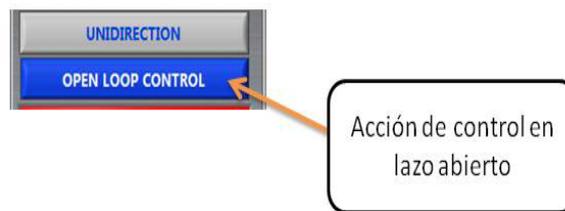


Figura 76. Botón para la elección de modos UAV control en lazo abierto

La acción de control en lazo cerrado, al emplear las aplicaciones en modo de vuelo automático es imperativo utilizar el área de manejo de aplicación de control y trayectorias, para utilizar el modo automático es necesario pulsar los botones de “UAV Mode” como se aprecia en la Figura 77.



Figura 77. Botón para la elección de modo UAV para el control en lazo cerrado

Área de manejo de aplicación de control y trayectorias, esta área está enfocada al proyecto, sirve para programar las aplicaciones de control que se puede implementar el control de posición del AR.Drone 2.0.

Una vez iniciado el enlace de comunicación y los sistemas de navegación establecidos y listos para la acción de vuelo mediante el área de inicialización, es en este punto donde esta área

entra en práctica utilizando la zona de panel de instrumentos de navegación mostrada en la Figura 78.



Figura 78. Entorno de instrumentos de navegación

Las zonas dependen mucho del modo de vuelo que se escoja dependiendo de la utilización que se requiera. Si se tiene presente un vuelo manual entonces se emplea la zona de panel de instrumentación de navegación, con el fin de visualizar los parámetros que se obtiene en el momento de vuelo del cuadricóptero mostrada en la Figura 79, se describe cada uno de estos parámetros:

- ❖ **Velocidad del cuadricóptero**, indicador que presenta la velocidad del AR.Drone 2.0 en vuelo horizontal, determinado en unidades m/seg,
- ❖ **Horizonte móvil**, este indicador presenta una proyección de movimiento y variación de los ángulos Roll y Pitch.
- ❖ **Altitud**, determinado por el sensor ultrasónico, este indicador presenta la altura que el cuadricóptero puede subir hasta un determinado nivel de altitud.
- ❖ **Brújula geostacionaria**, este indicador presentar una estimación de posición geostacionario con referencia al norte magnético

CAPÍTULO 5

- ❖ **Nivel de batería**, un parámetro muy importante en el sistema del cuadricóptero, este indicador visualiza el nivel de consumo de energía de la batería, en el transcurso del tiempo de vuelo.

El usuario tiene la posibilidad de controlar el cuadricóptero de dos opciones de programación, la primera opción es cuando se aplica un control en lazo abierto indicando su trayectoria, su movimiento en uno o en dos ejes, y lo más importante el manejo de los datos de navegación que el cuadricóptero genera en el instante de vuelo, existen tres sectores que se detalla en la Figura 79.



Figura 79. Zona de configuración de control

Entradas de control en lazo abierto “Open Loop Input Pattern”, comprende los ingresos de las variables de los ángulos Pitch y Roll, para cargar estas entradas se tiene que pulsar el botón “Load Input”, como se muestra en la Figura 80.

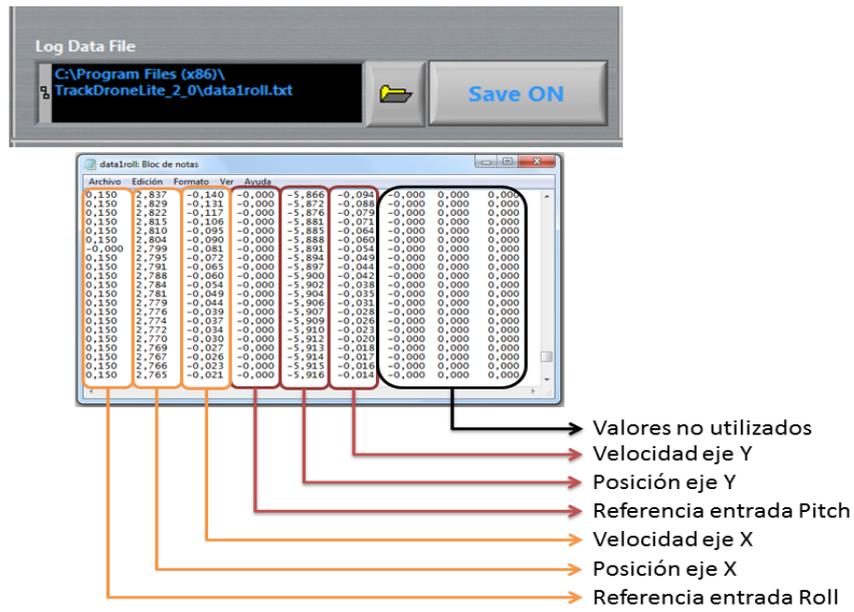


Figura 82. Respuesta del vuelo, archivo generado

La Figura 83 muestra el ingreso de los parámetros de control PID de las posiciones (x, y), determinados en el capítulo 4 también denominada como “Controls Parameters”.

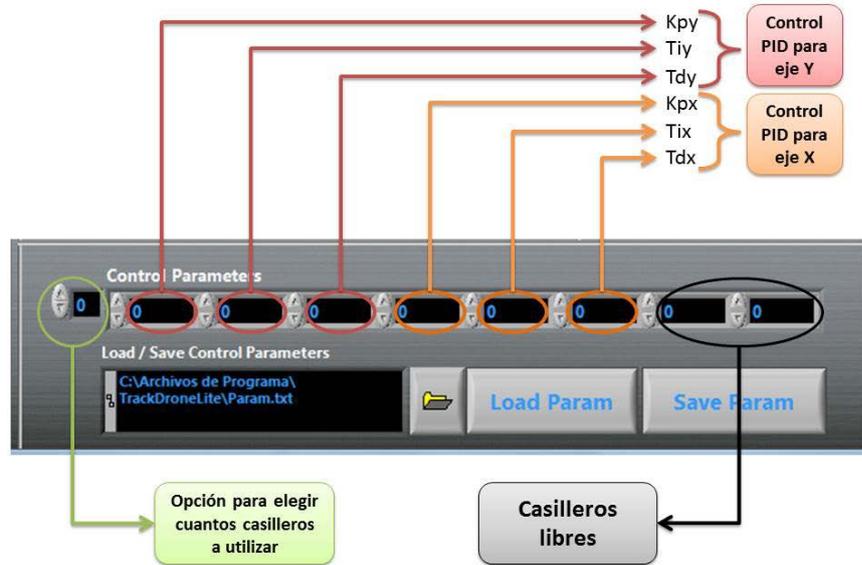


Figura 83. Ingreso de los parámetros de control PID

TrackDrone Lite 2.0 tiene una opción de cargar este archivo automáticamente como se aprecia en la Figura 84, sin la necesidad de ingresar las variables de modo manual.

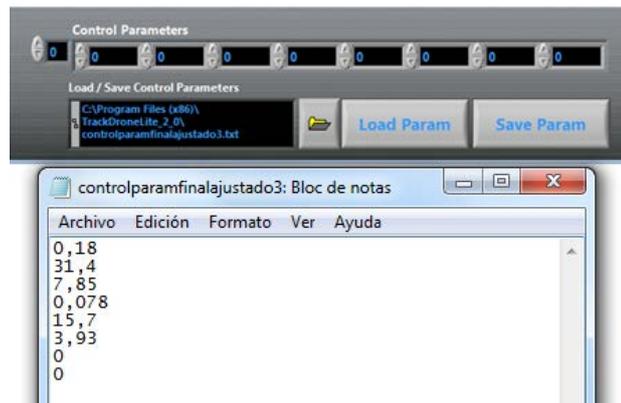


Figura 84. Zona que carga el archivo .txt, parámetros de control

Área de visualización de posición, presenta la visualización del campo de navegación que el vehículo tiene como desplazamiento en los ejes (x, y), cualquier modo de vuelo que se elija ya sea modo manual o automático, el programa siempre estima la trayectoria que recorra el cuadricóptero. Al existir un desplazamiento el cuadricóptero se lo representa a través de una figura en forma de X de color rojo, la trayectoria que genera el cuadricóptero se lo representa con una línea continua de color verde como se aprecia en la Figura 85. Para borrar estos desplazamientos realizados existe un botón que reinicia todos estos trazos para dejar limpio el campo, esto se lo hace pulsando el botón “Reset Track”.



Figura 85. Zona de visualización de desplazamientos del cuadricóptero

CAPÍTULO 5

El punto donde despegue el cuadricóptero está en la coordenada (0, 0), pero si se necesita el cambio de coordenada de despegue en otro lugar dentro del campo de trabajo, entonces se necesita pulsar el botón “Reset Position” que se aprecia en la Figura 86, con la intención que genere otra coordenada nueva de despegue del cuadricóptero.

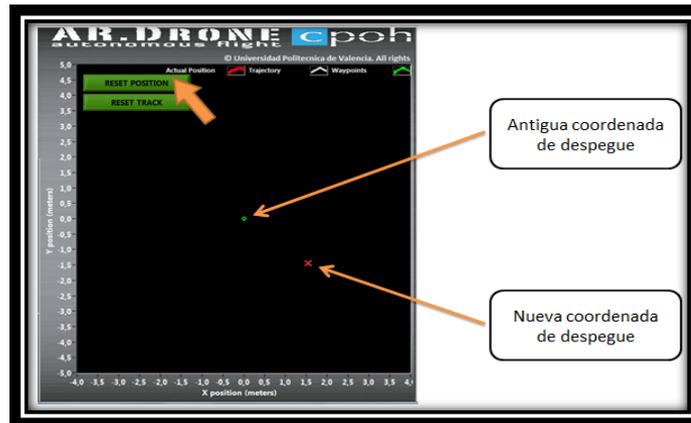


Figura 86. Entorno de desplazamientos del cuadricóptero

La interacción entre herramientas de software; específicamente la plataforma de programación visual C++ 2010 e interfaz gráfica TrackDrone Lite 2.0; mediante las técnicas y estrategias de control aplicados al diseño, se consigue el objetivo de la comprobación y ejecución de los procesos y acciones planteados para la puesta en práctica del sistema de control de posición en el cuadricóptero, en la Figura 87 se visualiza el concepto global de la correlación del sistema de programación con el entorno físico.

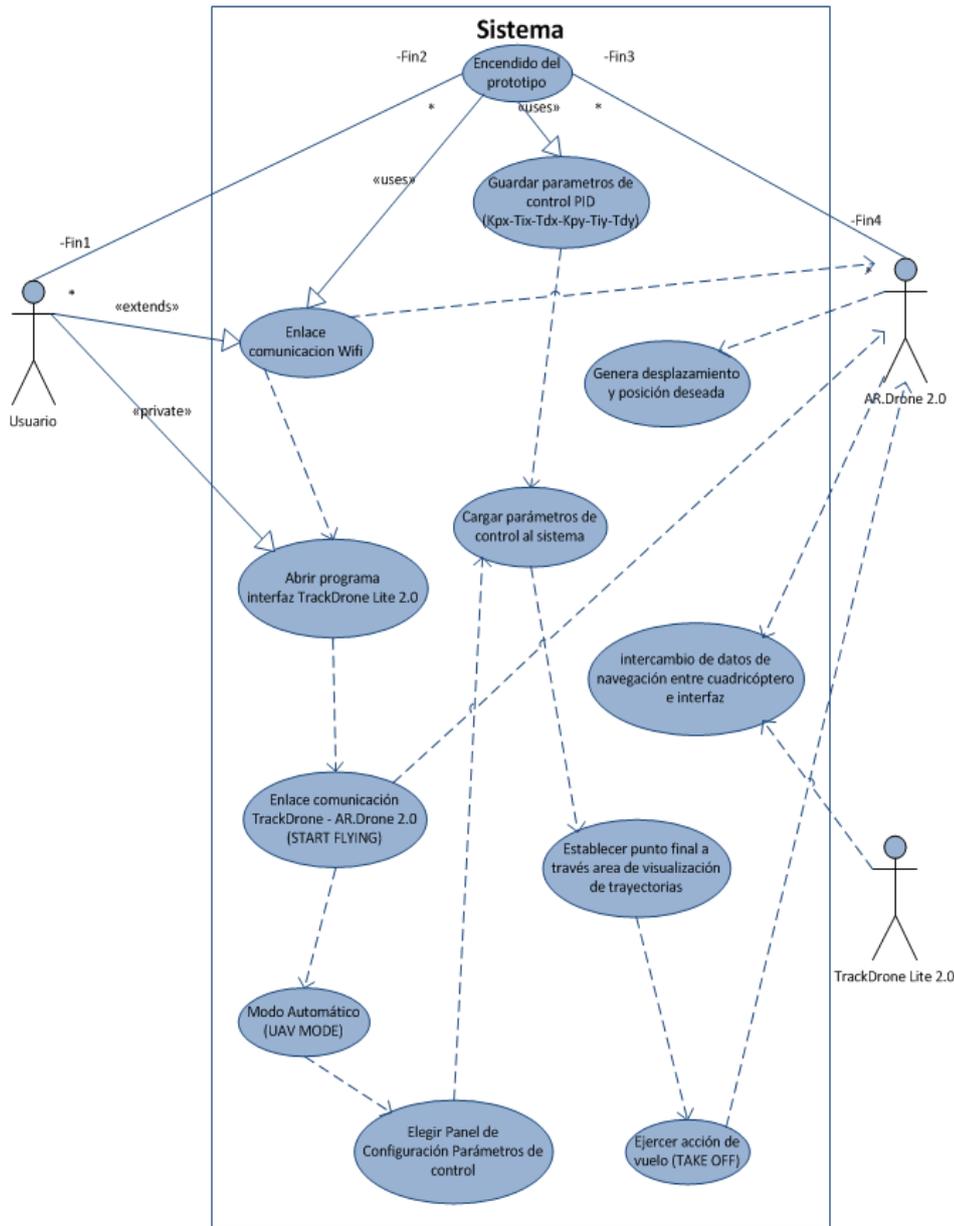


Figura 87. Diagrama de casos de usos aplicado a la metodología (Usuario – Sistema – Cuadricóptero)

5.2 Programación del sistema de control de posición

El presente capítulo proporciona un análisis sustancial de la estructura de programación tales como, la utilización del lenguaje de programación asociando los parámetros de control expuestos en el capítulo 4, vincular la algoritmia diseñada del sistema de control de posición del cuadricóptero con una interfaz gráfica. Estas técnicas se establecen como objetivo el

CAPÍTULO 5

plantear una de las diferentes soluciones que existe para la verificación visual de modo real del sistema de control planteado para el posicionamiento del AR.Drone 2.0.

La Figura 88 muestra la forma de trabajo entre el TrackDrone lite 2.0 y el AR.Drone 2.0, que mantiene una constante comunicación, el cual comparten información necesaria e importante para el sistema de control aplicado al cuadricóptero.

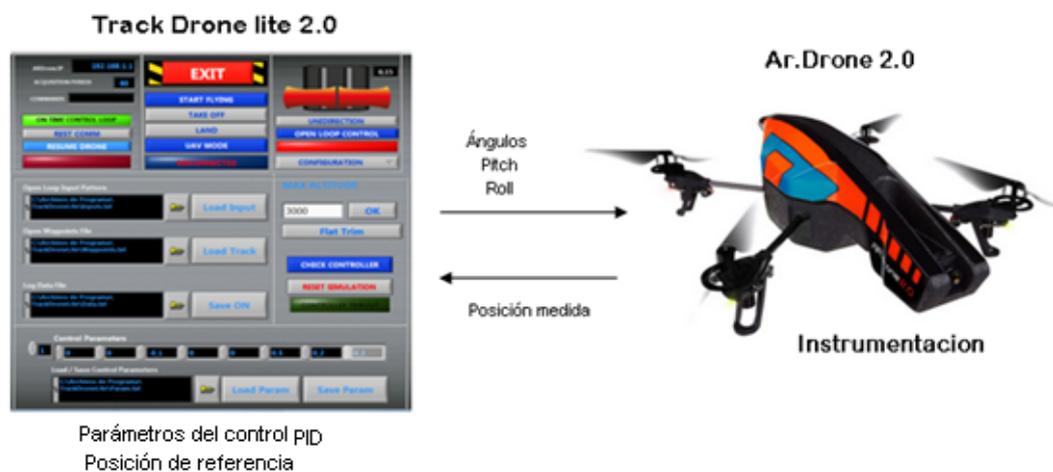


Figura 88. Implementación del diseño de control

El lenguaje de programación utilizado para el código es C++, proclamado como el lenguaje ideal para la creación del código del diseño de control, una herramienta importante del TrackDrone Lite 2.0, para la ejecución del algoritmo de control, es que permite alojar toda la estructura del código generado a través de una biblioteca de enlace dinámico privada o archivo DLL³⁰, como también la reutilización del código para posteriores ajustes, en la Figura 89, se aprecia el conjunto de la “ley de control PID”, programado en lenguaje C++, el cual contempla el llamado de los parámetros del control de posición (x, y), como también la lectura de las posiciones inicial y final en tiempo real.

³⁰ DLL biblioteca de enlace dinámico por sus siglas en inglés Dynamic-link library



Figura 89. Archivo con extensión .dll

Para generar el archivo .dll se apoya con la asistencia del paquete de plataformas de programación visual studio 2010 (Microsoft®, 2011), específicamente con la ayuda del programa Microsoft visual C++ 2010.

El siguiente paso consiste en situar este archivo dentro de la carpeta donde se instaló el directorio principal del TrackDrone Lite 2.0 en el disco duro, la dirección donde va alojado el archivo ejecutable .dll se especifica en la Figura 90.

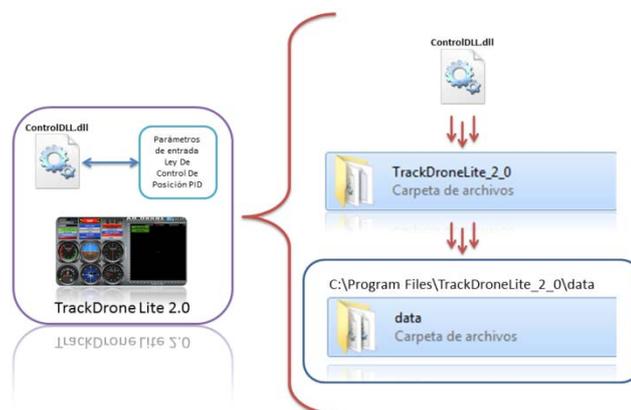


Figura 90. Lugar y dirección en donde va archivado "ControlDLL.dll"

La inicialización de parámetros y la continua lectura de las posiciones del cuadricóptero son a través del programa TrackDrone Lite 2.0, a través de punteros estos parámetros de control establecen prioridad importante en el algoritmo de programación. Con un profundo análisis el programa se aclara con el diagrama de actividades presentando por la Figura 91.

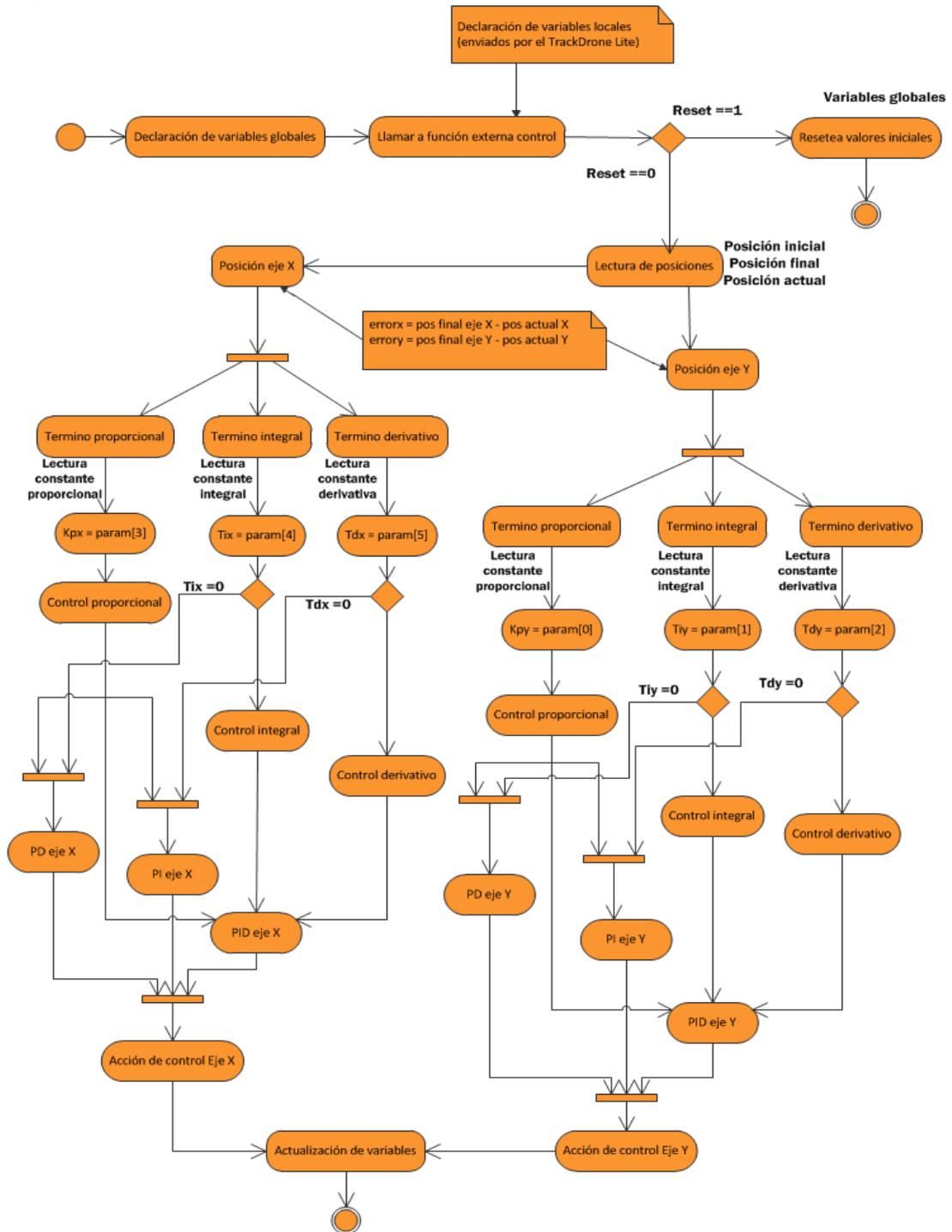


Figura 91. Diagrama UML de actividades de la algoritmia de programación.

El diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema. En UML, una clase es representada por un rectángulo que posee tres divisiones, y está compuesto por los siguientes elementos:

CAPÍTULO 5

- Clase: atributos, métodos y visibilidad.

En la Figura 92, se exhibe el diagrama de clases con la integración de vista estática del sistema. Se define las características de métodos y atributos contemplados en la programación.

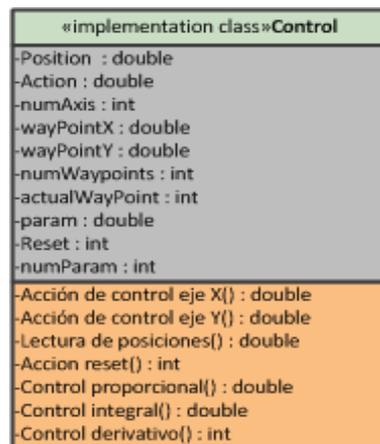


Figura 92. Diagrama UML de clases muestran los atributos y operaciones de la clase Control.

La estrategia de programación plantea una estructura dinámica de datos en base a la posición final deseada y a los parámetros de control, actualizar sus estados y/o datos por nuevos valores, por lo tanto el algoritmo se divide en tres bloques:

- ❖ Bloque de ingreso,
- ❖ Bloque de control de posición,
- ❖ Bloque de realimentación.

El bloque de ingreso administra las variables locales de la posición actual del cuadricóptero llamadas desde la función de control, se configuran como operadores para la aritmética en función a las distancias entre el punto de origen hasta al punto final deseado, mostrada en la Figura 93 y con una descripción específica en la tabla 9, se declaran variables para los ejes (x, y).

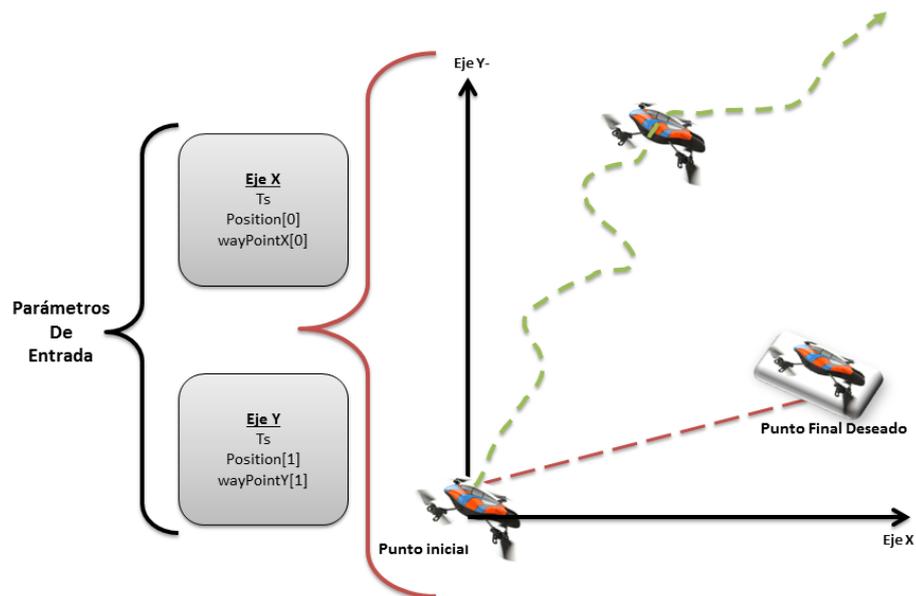


Figura 93. Bloque de dirección programada del TrackDrone Lite 2.0

Tabla 9: Descripción de los parámetros y variables que intervienen en el bloque de ingreso.

<u>Variable</u>	<u>Tipo</u>	<u>Descripción</u>
Ts	Double	Tiempo de muestreo, cada 60 mseg
Position[0]	Double	Posición actual del cuadricóptero en eje X
Position[1]	Double	Posición actual del cuadricóptero en eje Y
wayPointX[0]	Double	Vector dimensión, contiene componente X del punto final deseado hacia donde tiene que llegar el cuadricóptero
wayPointY[0]	Double	Vector dimensión, contiene componente Y del punto final deseado hacia donde tiene que llegar el cuadricóptero

Elaborado por: Autores del proyecto

El segundo bloque contempla las funciones de operación matemática que representan las leyes de control que exhibe la estructura PID, utilizando los parámetros de entrada a través del TrackDrone Lite 2.0 que se muestra en la Figura 94 y en detalle en la tabla 10, el control presenta una estabilidad ocasional llegando al punto final deseado por parte del usuario, sin embargo no crea un acción de control en lazo cerrado ya que su vuelo es infinito.

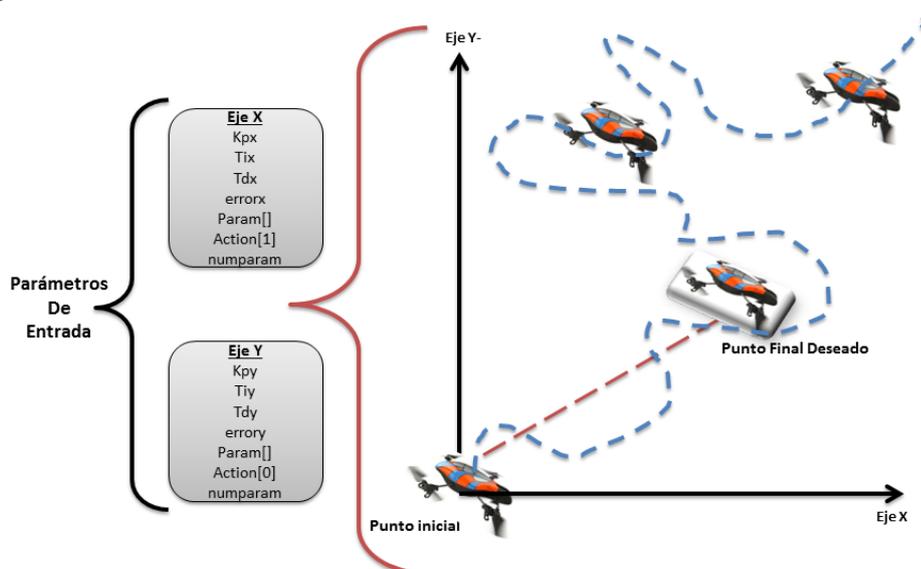


Figura 94. Esquema de programación con ley de control PID

Tabla 10: Descripción de los parámetros y variables que intervienen en el bloque de lazo abierto.

<u>Variable</u>	<u>Tipo</u>	<u>Descripción</u>
Kpx	Double	Constante proporcional para el eje X
Kpy	Double	Constante proporcional para el eje Y
Tix	Double	Tiempo integral para el eje X
Tiy	Double	Tiempo integral para el eje Y
Tdx	Double	Tiempo derivativo para el eje X
Tdy	Double	Tiempo derivativo para el eje Y
errorx	Double	Residuo del valor de la posición final menos la posición inicial para el eje X
errory	Double	Residuo del valor de la posición final menos la posición inicial para el eje Y
numparam	Int	Número de parámetros a utilizar para las acciones de control, son 6 parámetros dispuestos para los dos ejes
Param[]	Double	Es un vector dimensión, y se encuentra a disposición del usuario para poder transmitir parámetros entre la interfaz del TrackDrone Lite con el archivo .dll
Action[0]	Double	Acción de control donde se aloja el control clásico PID del cuadricóptero para el eje X
Action[1]	Double	Acción de control donde se aloja el control clásico PID del cuadricóptero para el eje Y

Elaborado por: Autores del proyecto

El bloque de realimentación actualiza la información de las distancias con respecto a la posición del cuadricóptero en tiempo real, conforme llega a su punto de destino el error tiende

a disminuir y por ende el AR.Drone 2.0 se mantiene de forma continua en el punto final, esta información se muestra en la Figura 95, y las variables que intervienen se precisa en la tabla 11.

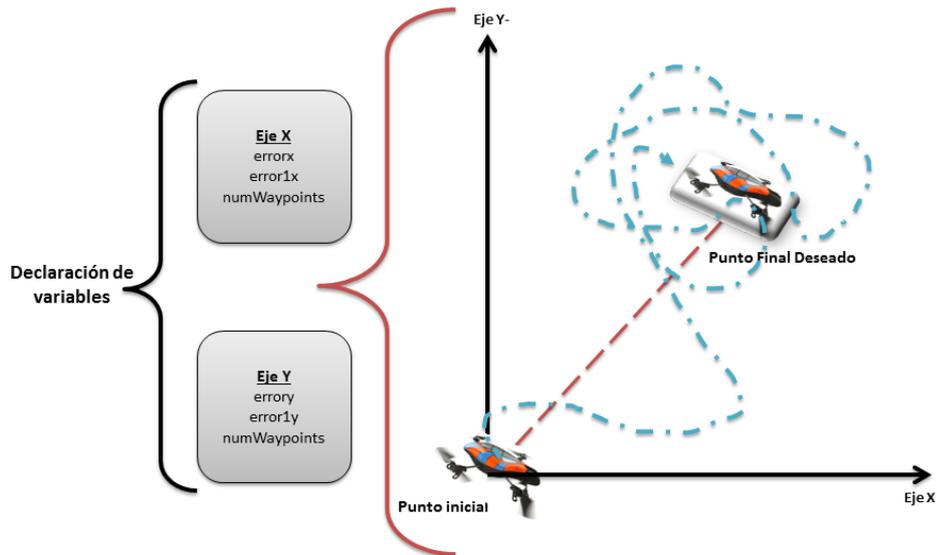


Figura 95. Esquema del programa realimentado para la acción de control PID

Tabla 11: Descripción de las variables que intervienen en el bloque de lazo cerrado.

<u>Variable</u>	<u>Tipo</u>	<u>Descripción</u>
error1x	Double	Error actualizado para posterior realimentación para el eje X
error1y	Double	Error actualizado para posterior realimentación para el eje Y
numwayPoints	Int	Número de puntos finales, en este caso solo existe un punto final ya que el control diseñado es aplicable como regulador

Elaborado por: Autores del proyecto

CAPÍTULO 6

PRUEBAS Y RESULTADOS

6.1 Ensayos y pruebas realizados

Con la experimentación efectuada en los lugares exactos, el sistema de control aplicado al cuadricóptero se comporta de manera distinta al utilizar un ordenador con un procesador dual Core, en comparación con un ordenador de procesador Core I5, las diferencias fueron evidentes en la comunicación, tiempo real, envío y recepción de datos. En este capítulo se aborda los experimentos realizados y del control de posición aplicado al prototipo.

Los lugares que se considera para el desarrollo de las pruebas son a 500m sobre el nivel del mar, ya que el modelo hallado funciona a esa altura, como se determina en el capítulo 3. La experimentación es outdoor³¹.

En el instante de despegue, el cuadricóptero controla su posición inicial mediante la algoritmia de control diseñado para alcanzar el punto final deseado, como se muestra en la Figura 96.

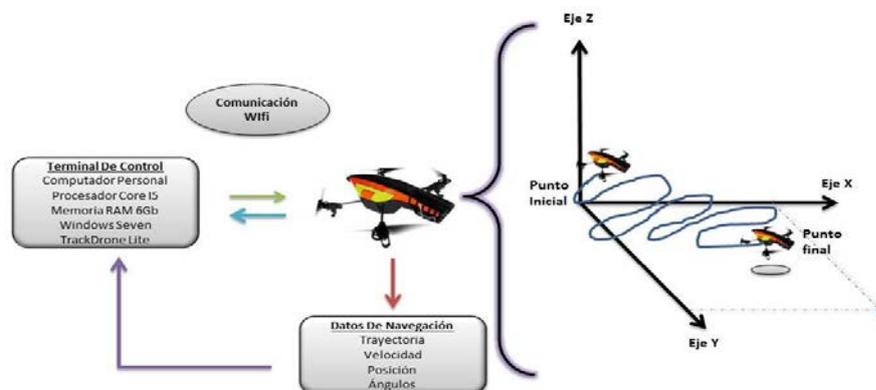


Figura 96. Planificación de experimentos del sistema de control del cuadricóptero

³¹ OUTDOOR Palabra que refleja algún procedimiento que se realiza al aire libre.

CAPÍTULO 6

Descripción y parámetros geográficos del lugar de experimentación

La tabla 12, muestra las propiedades geográficas de la zona de pruebas que se consideró para la aplicación del sistema de control diseñado para el vuelo del cuadricóptero,

Tanto la magnitud de la velocidad del viento como su dirección es un factor importante ya que presenta una perturbación que afecta al sistema de control, existe una perturbación dentro del sistema. El diseño del control prevé que al existir dichas perturbaciones el cuadricóptero logre llegar a un punto con una exactitud congruente.

Tabla 12. Parámetros técnicos y geográficos del lugar de experimentación

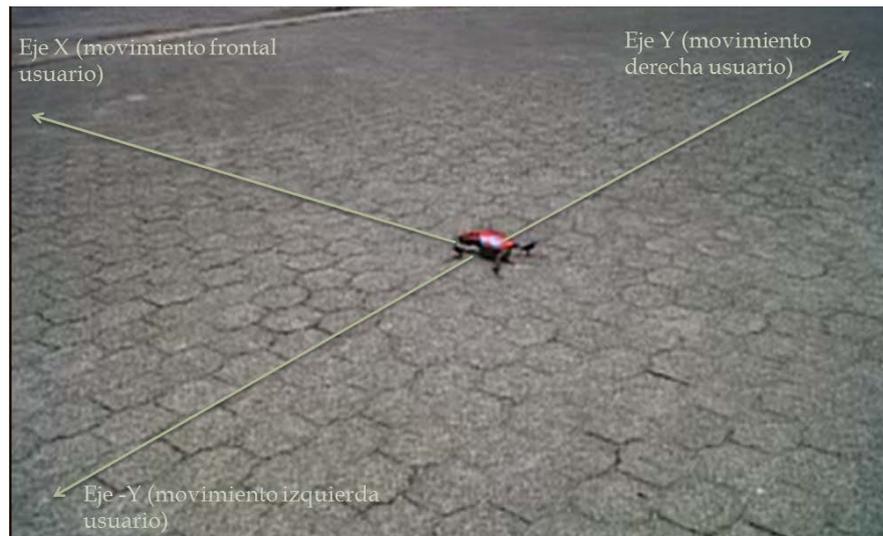
Lugar De Ensayos	Parque De La Juventud - Santo Domingo
Altitud Geográfica	589 [m]
Temperatura Promedio	30 grados centígrados
Punto Geográfico	0°14'51.24'' S 79°08'46.87'' O
Presión Atmosférica	104Kpa
Humedad	70 [%]
Estilo De Vuelo	Outdoor
Velocidad De Viento	15 [Km/h]
Dirección De Viento	N con dirección S

Elaborado por: Autores del proyecto

Experimentación;

En la fotografía 10 se representa el área donde se realiza las pruebas del control de posición, tomando como referencia los ejes de coordenadas y el sentido de dirección de la cámara del cuadricóptero, son importantes para el momento del análisis matemático y de estadística de vuelo, se mantiene como norma fundamental a respetar la regla de la mano derecha que indica los ejes propios para el dimensionamiento de ejes de las pruebas realizadas.

CAPÍTULO 6



Fotografía 10. Área de experimentación del sistema de control del cuadricóptero

Para el movimiento hacia adelante donde se encuentra la cámara de video se localiza el eje X positivo, en contraposición el movimiento hacia atrás se localiza el eje X negativo, el traslado del cuadricóptero hacia el costado derecho se tiene el eje Y positivo, en contraparte el costado izquierdo se encuentra el eje Y negativo.

En la Figura 97 se visualiza los experimentos establecidos para el control de posición del cuadricóptero visualizando de forma clara las coordenadas para el cual el cuadricóptero se dirige al punto final. Por cada experimento realizado toma el nombre de EXP1 y EXP2, se realiza dos experimentos aplicando el control de posición en el plano de coordenadas (x, y). El análisis desarrollado para cada uno de las pruebas de control de posición interpreta el comportamiento del tiempo de respuesta de la planta y del control PID, la velocidad de ejecución de la comunicación entre el computador con el cuadricóptero.

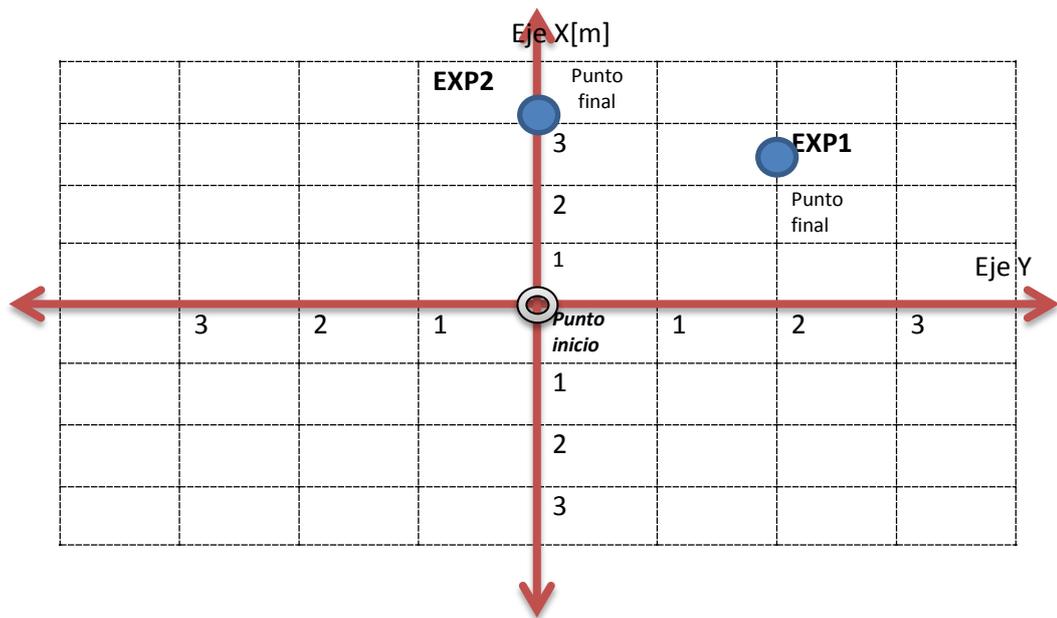


Figura 97. Mapa de coordenadas para las pruebas del control de posición EXP1, 2 del cuadricóptero

Experimentación EXP1

Tabla 13. Características del experimento EXP1

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	2.5
Eje Y	0	2

Elaborado por: Autores del proyecto

El desplazamiento del cuadricóptero se lo representa en la Figura 98, mediante la herramienta Matlab y tomando los valores que genera el AR.Drone 2.0 en el instante de vuelo, se logra representar el desplazamiento desde el punto inicial hasta el punto final de las coordenadas de EXP1.

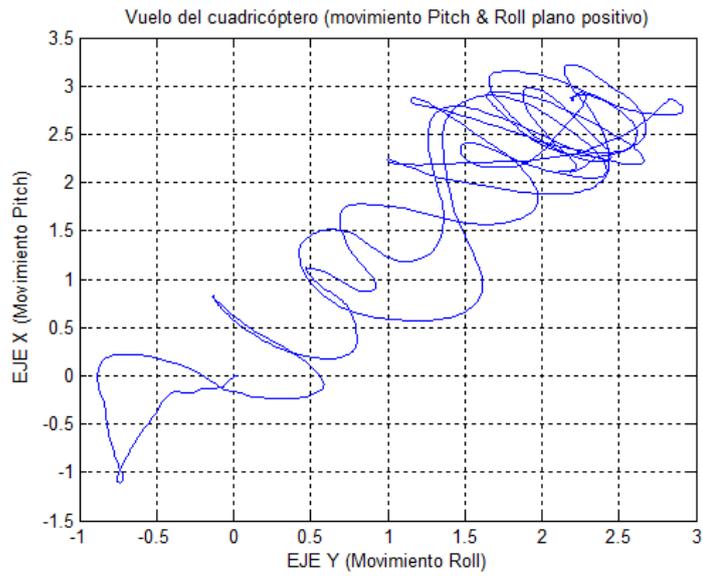


Figura 98. Desplazamiento en Matlab, del cuadricóptero hacia el punto final deseado EXP1

La fotografía 11 representa la forma real del vuelo descrito por el cuadricóptero hacia el punto deseado.



Fotografía 11. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento EXP1

La Figura 99 muestra el comportamiento de la posición del eje y, con respecto al tiempo, como indica en la tabla 20 el punto inicial y el punto final.

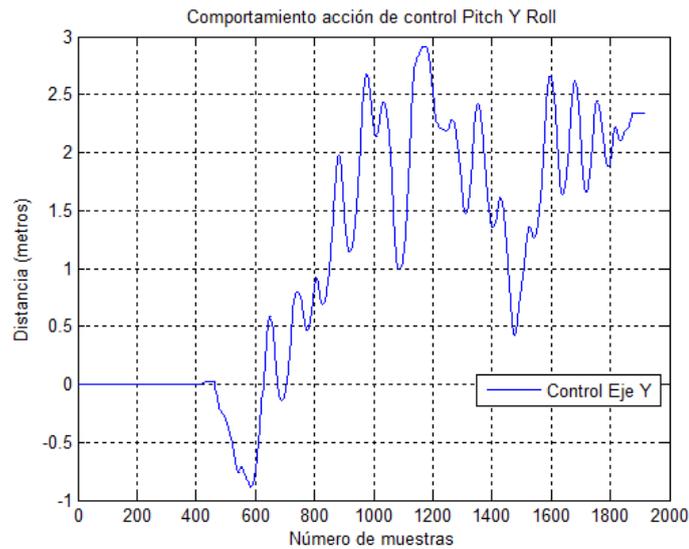


Figura 99. Respuesta en el tiempo del eje y, EXP1.

Una muestra representa la información o iteración generada por el cuadricóptero en el transcurso del tiempo, cada tiempo de muestreo es de 60ms, es así que dependiendo el número de muestras se calcula el tiempo de prueba de vuelo, en el siguiente apartado se detalla el cálculo de los tiempos que se registran en el momento de la experimentación con el cuadricóptero. Se toma como referencia el experimento EXP1, cuando el cuadricóptero realiza el vuelo desde la coordenada inicial hasta el punto 2m a lo largo del eje `x`, en adelante esta metodología será la herramienta de uso para el cálculo de exactitud, ya que es fundamental y necesario para los demás experimentos.

Número de muestras $\rightarrow m_{EXP1} = 1920$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación - EXP1 $\rightarrow t_{EXP1} = m_{EXP1} * t_s \Rightarrow 1920 * 0.06$

$$t_{EXP1} = 115.20[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 900 * 0.06$

$$T_{PIDY} = 54[s]$$

CAPÍTULO 6

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{\text{dead}} = 450 * 0.06$

$$T_{\text{dead}} = 27[\text{s}]$$

La Figura 100 muestra el comportamiento de la posición del eje x, con respecto al tiempo, como indica en la tabla 20 el punto inicial y el punto final.

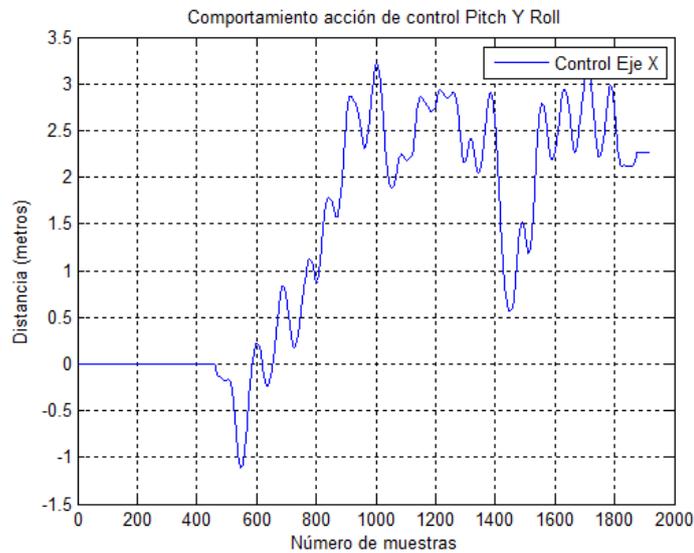


Figura 100. Respuesta en el tiempo del eje x, EXP1

Número de muestras $\rightarrow m_{\text{EXP1}} = 1920$

Tiempo de muestreo $\rightarrow t_s = 0.06[\text{s}]$

Tiempo de experimentación - EXP1 $\rightarrow t_{\text{EXP1}} = m_{\text{EXP1}} * t_s \Rightarrow 1920 * 0.06$

$$t_{\text{EXP1}} = 115.20[\text{s}]$$

Tiempo de estabilización eje X $\rightarrow T_{\text{PIDX}} = 1100 * 0.06$

$$T_{\text{PIDX}} = 66[\text{s}]$$

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{\text{dead}} = 450 * 0.06$

$$T_{\text{dead}} = 27[\text{s}]$$

Tabla 14 Respuesta del sistema de control EXP1

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	66	20
Eje Y	54	25

Elaborado por: Autores del proyecto

Experimentación EXP2

Tabla 15. Características del experimento EXP1

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	3
Eje Y	0	0

Elaborado por: Autores del proyecto

En la Figura 101, muestra la dirección del vuelo del cuadricóptero mediante la herramienta de Matlab, a través del eje x, representando el desplazamiento desde el punto inicial hasta el punto final de las coordenadas del EXP2.

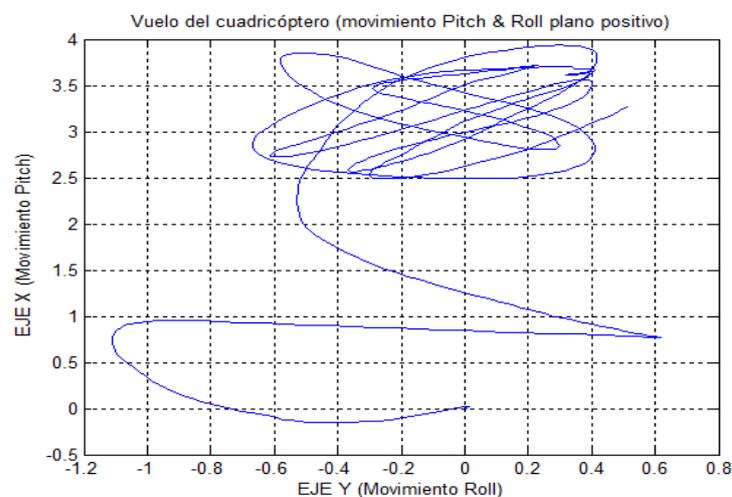
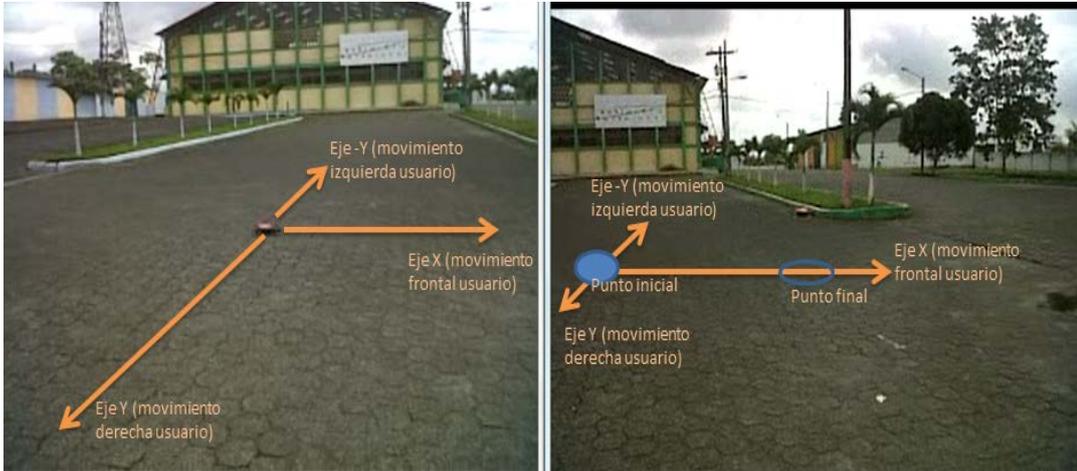


Figura 101. Gráfica del desplazamiento en Matlab, del cuadricóptero hacia el punto final deseado EXP2

CAPÍTULO 6

En la fotografía 12, representa de forma real el vuelo del cuadricóptero descrito hacia el punto final.



Fotografía 12. Vuelo real del cuadricóptero desde el punto inicial hacia el punto final del experimento EXP2

La Figura 102 muestra el comportamiento de la posición del eje y, con respecto al tiempo.

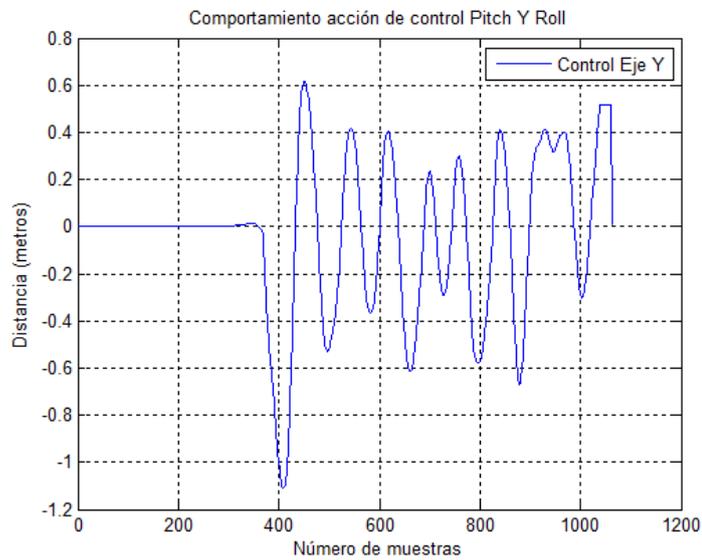


Figura 102. Respuesta en el tiempo del eje y, EXP2.

Número de muestras $\rightarrow m_{EXP2} = 1061$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – EXP2 $\rightarrow t_{EXP2} = m_{EXP2} * t_s \Rightarrow 1061 * 0.06$

CAPÍTULO 6

$$t_{EXP2} = 63.66[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 410 * 0.06$

$$T_{PIDY} = 24.60[s]$$

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{dead} = 380 * 0.06$

$$T_{dead} = 22.80[s]$$

La Figura 103 muestra el comportamiento de la posición del eje x, con respecto al tiempo.

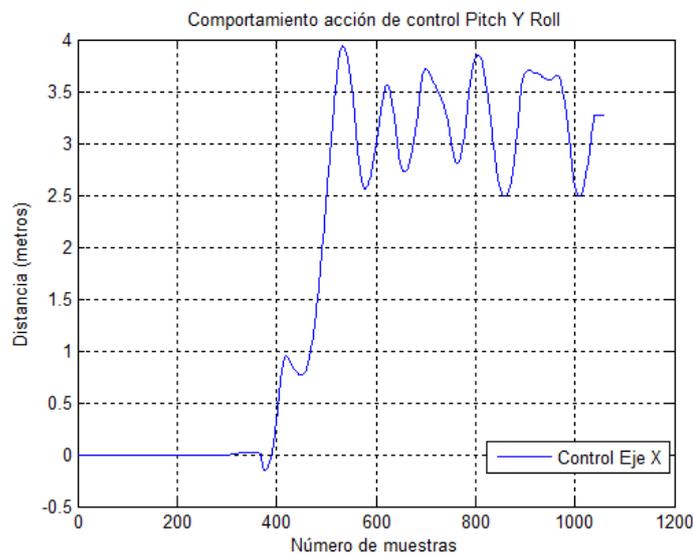


Figura 103. Respuesta en el tiempo del eje x EXP2

Número de muestras $\rightarrow m_{EXP2} = 1920$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – EXP2 $\rightarrow t_{EXP2} = m_{EXP2} * t_s \Rightarrow 1920 * 0.06$

$$t_{EXP2} = 115.20[s]$$

Tiempo de estabilización eje X $\rightarrow T_{PIDX} = 580 * 0.06$

$$T_{PIDX} = 34.80[s]$$

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{dead} = 380 * 0.06$

$$T_{\text{dead}} = 22.8[\text{s}]$$

Tabla 16. Respuesta del sistema de control EXP2

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	66	20
Eje Y	54	25

Elaborado por: Autores del proyecto

Exactitud y precisión

Este apartado desglosa de manera sustancial la exactitud y precisión realizada por el cuadricóptero una vez que se ejecutó los experimentos EXP1 y EXP2, el estudio de la exactitud representa el porcentaje de error que obtuvo el vehículo al llegar al punto deseado, mostrado en la Gráfica 1.

La exactitud se representa como el término usado para describir la cercanía con la cual el valor medido se aproxima al valor real de la variable que está siendo medida (Ortiz T, 1998), se relaciona de manera directa con el cálculo del error relativo ya que considera la media aritmética del valor medido semejante al valor ideal, y el resultado se simboliza como el porcentaje de proximidad. Para estos casos puntuales se utiliza todas las muestras generadas en el transcurso del tiempo de experimentación, se suma todos los componentes y se divide para el número total de muestras.

Un ejemplo tácito se expone de la siguiente manera y aplicado de forma instructiva con el cálculo de la exactitud del experimento EXP1 en el eje `y`, esta metodología de cálculo se aplica para los posteriores experimentos realizados a futuro.

Número de muestras $\rightarrow m_{EXP1} = 1920$

Valor deseado $\rightarrow V_{EXP1} = 2[\text{m}]$

CAPÍTULO 6

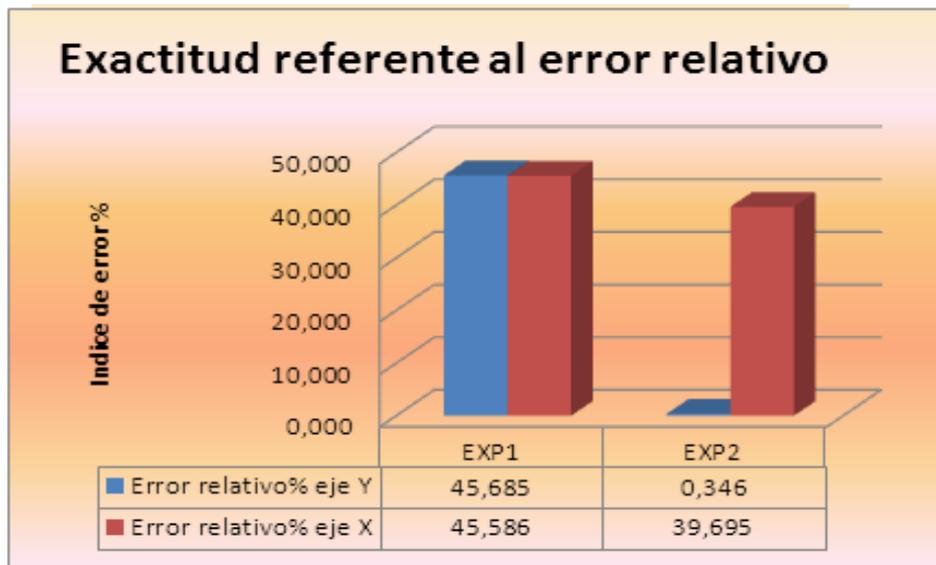
Media aritmética con 1920 muestras $\rightarrow \bar{x} = \frac{\sum x_{1920}^1}{1920} \Rightarrow \bar{x} = 1.086[m]$

Error absoluto $\rightarrow e_{\text{abs}} = |V_{\text{EXP1}} - \bar{x}| \Rightarrow |2 - 1.086|$

$$e_{\text{abs}} = 0.9137$$

Error relativo $\rightarrow e_{\text{rel}} = \frac{e_{\text{abs}}}{V_{\text{EXP1}}} * 100 \Rightarrow \frac{0.9137}{2} * 100$

$$e_{\text{rel}} = 45.69\%$$



Gráfica 1. Exactitud de la posición para los experimentos EXP1 y EXP2.

El estudio de la precisión mantiene un criterio de discernir la cantidad de distancia que se desvía el cuadricóptero al llegar al punto final, es expresado en metros y es la distancia medida que corresponde al diámetro del margen o circunferencia descrita por el vuelo del AR.Drone 2.0, este valor se asocia con la desviación estándar, término que interpreta las medidas de dispersión que el cuadricóptero genera alrededor del punto final, se expone los resultados en la Gráfica 2 realizada por los experimentos EXP1, EXP2.

CAPÍTULO 6

La estructura matemática que calcula la precisión se expone a continuación a través del experimento EXP1 en el eje Y, y el mismo procedimiento será aplicado a los consecuentes pruebas requeridas.

Número de muestras $\rightarrow m_{EXP1} = 1920$

$$\text{Desviación estándar} \rightarrow \sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n}} \Rightarrow \sqrt{\frac{316.95}{1920}}$$

$$\sigma = 1.0627[\text{m}]$$



Gráfica 2. Precisión de la posición para los experimentos EXP1 y EXP2

6.2 Ajustes del sistema

Para estas nuevas pruebas de campo, los parámetros del lugar geográfico varían en consideración a la altura y la dirección del aire, comparando la tabla 12 y la tabla 17. Al tener una visión sobre el comportamiento del cuadricóptero y el control aplicado sobre el mismo por las pruebas realizadas EXP1 y EXP2, se mejora el hardware del cuadricóptero el cual se mantiene en condiciones desfavorables a las originales, esto es debido a las constantes

pruebas rigurosas que ha sufrido el cuadricóptero en los instantes de modelamiento y pruebas de control iniciales, los materiales que se cambiaron son; carcasa de protección, las hélices, los motores y los piñones, prácticamente el sistema de vuelo, además se ajusta los parámetros del controlador mostrados en el capítulo 4, sección 4.1.4.

Tabla 17. Parámetros técnicos y geográficos del lugar de experimentación final

Parámetros	Valores y datos
Lugar De Ensayos	ESPE – IASA II Sede Santo Domingo, instalaciones administrativas
Altitud Geográfica	223 [m]
Temperatura Promedio	29 grados centígrados
Punto Geográfico	0°30'00.21'' S 79°20'54.28'' O
Presión Atmosférica	114Kpa
Humedad	73 [%]
Estilo De Vuelo	Outdoor
Velocidad De Viento	17 [Km/h]
Dirección De Viento	N con dirección O

Elaborado por: Autores del proyecto

En la fotografía 13, presenta el área para desarrollar los experimentos que se realiza el control de posición del cuadricóptero visualizando de forma clara el sistema de referencia de las coordenada para cada eje, se coloca una cuerda color azul, el punto de inicio donde despega el cuadricóptero se simboliza con una lámina de color blanco, y cada unidad de medida es 50 [cm], que está marcada con cinta aislante color blanco.



Fotografía 13. Área para realizar el control de posición

Experimentación

En esta instancia de nueva experimentación, cada prueba de campo realizado toma el nombre de POS1, 2, 3, 4, 5, 6, se realiza seis experimentos comenzando desde el plano positivo Y-X, y se toma los puntos de prueba en consideración al movimiento anti horario, de esta manera la consigna es distribuir el ambiente de experimentación a lo largo de todo el sistema de coordenadas, tal y como se muestra en la Figura 104.

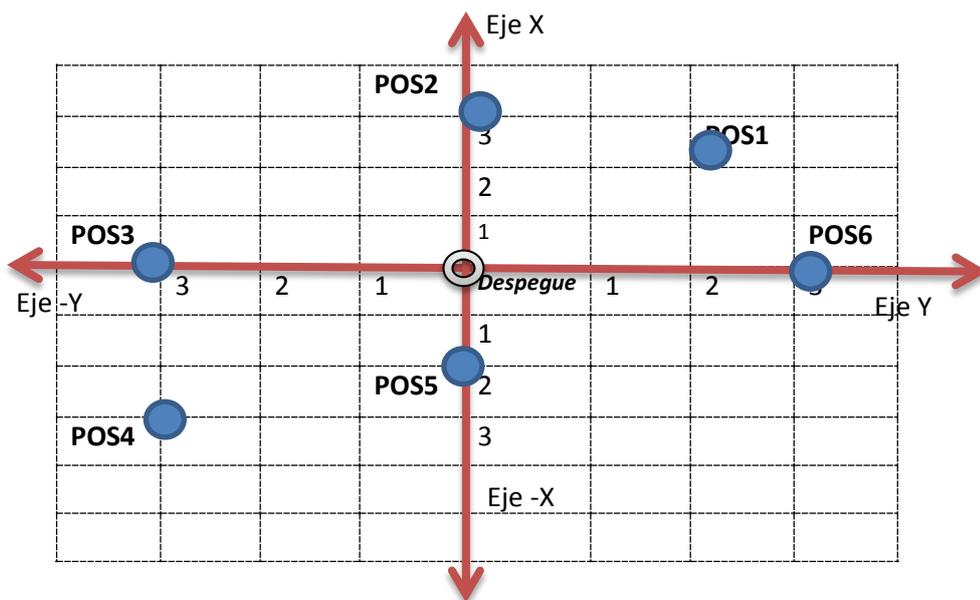


Figura 104. Mapa de coordenadas para las pruebas POS del control de posición del cuadricóptero

CAPÍTULO 6

Experimentación POS1

Características de POS1

Tabla 18. Características del experimento POS1

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	2,5
Eje Y	0	2

Elaborado por: Autores del proyecto

Desplazamiento del cuadricóptero hacia el punto final

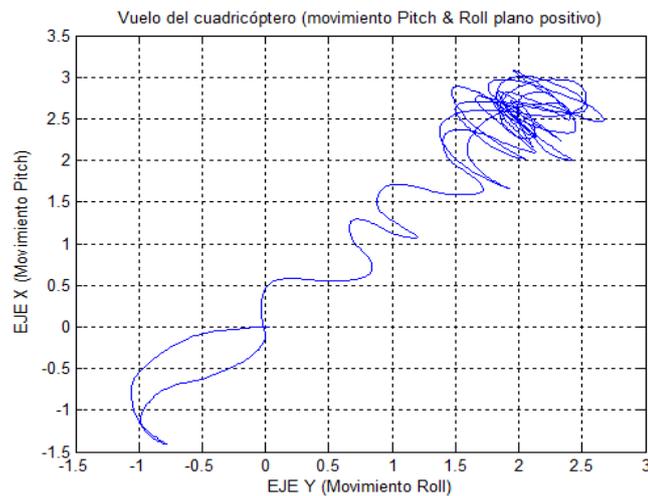


Figura 105. Desplazamiento en Matlab del cuadricóptero desde el punto inicial hacia el punto final del experimento POS1

La trayectoria descrita en color azul presenta una leve pérdida del control ya que el cuadricóptero en el instante de despegue tuvo una trayectoria negativa y por ende el desvío total del vehículo, hasta el instante de ejecución de acción de control que tomo su trayectoria hacia el punto final.

En la fotografía 14, representa la forma real del vuelo descrito por el cuadricóptero desde el punto inicial hacia el punto final deseado.



Fotografía 14. Vuelo real del cuadricóptero desde el punto inicial al punto final experimento POS1

Comportamiento de la posición del eje y con respecto al tiempo

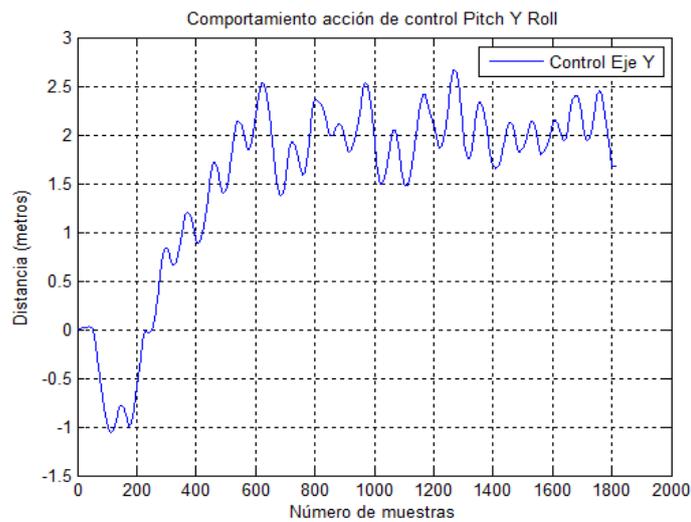


Figura 106. Respuesta en el tiempo del eje y, POS1

El tiempo total de experimentación es alrededor de los 110s, desde el tiempo de despegue al tiempo que llega al punto final. El tiempo de ejecución de la acción de control es de 33s, tiempo en que el cuadricóptero tiende a volar de manera errática hasta el instante t_{PID} , en adelante logra alcanzar el objetivo.

CAPÍTULO 6

Número de muestras $\rightarrow m_{POS1} = 1818$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS1 $\rightarrow t_{POS1} = m_{POS1} * t_s \Rightarrow 1818 * 0.06$

$$t_{POS1} = 109.08[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 550 * 0.06$

$$T_{PIDY} = 33[s]$$

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{dead} = 50 * 0.06$

$$T_{dead} = 3[s]$$

Comportamiento de la posición del eje x con respecto al tiempo.

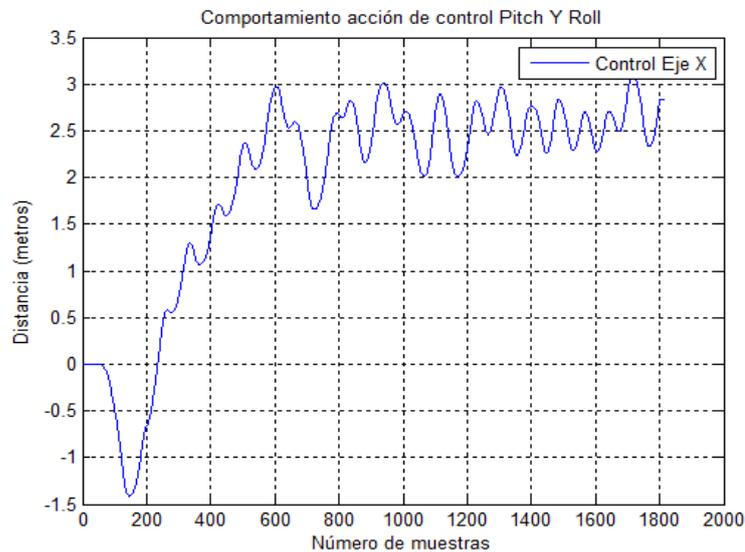


Figura 107. Respuesta en el tiempo del eje x, POS1

Número de muestras $\rightarrow m_{POS1} = 1818$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación - POS1 $\rightarrow t_{POS1} = m_{POS1} * t_s \Rightarrow 1818 * 0.06$

$$t_{POS1} = 109.08[s]$$

CAPÍTULO 6

Tiempo de estabilización eje X → $T_{PIDX} = 550 * 0.06$

$$T_{PIDX} = 33[s]$$

Tiempo muerto → es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición → $T_{dead} = 50 * 0.06$

$$T_{dead} = 3[s]$$

Tabla 19. Respuesta del sistema de control POS1

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	33	15
Eje Y	33	16

Elaborado por: Autores del proyecto

Experimentación POS2

Características de POS2

Tabla 20. Características del experimento POS2

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	3
Eje Y	0	0

Elaborado por: Autores del proyecto

Desplazamiento del cuadricóptero hacia el punto final

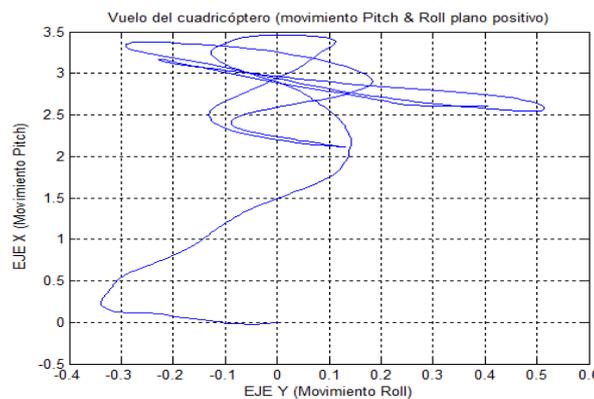


Figura 108. Desplazamiento en Matlab del cuadricóptero desde el punto inicial hacia el punto final del experimento POS2



Fotografía 15. Vuelo real del cuadricóptero desde el punto inicial al punto final experimento POS2

Comportamiento de la posición del eje y con respecto al tiempo

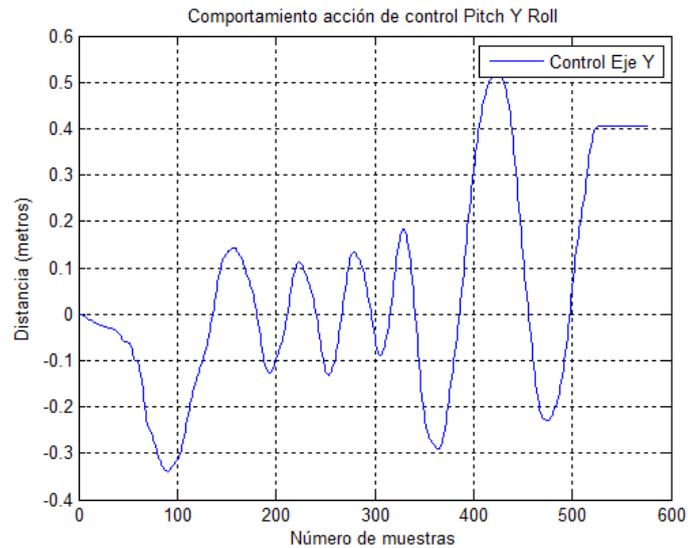


Figura 109. Respuesta en el tiempo del eje y, POS2

Número de muestras $\rightarrow m_{POS2} = 577$

Tiempo de muestreo $\rightarrow ts = 0.06[s]$

Tiempo de experimentación – POS2 $\rightarrow t_{POS2} = m_{POS2} * ts \Rightarrow 577 * 0.06$

$$t_{POS2} = 34.62[s]$$

CAPÍTULO 6

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 180 * 0.06$

$$T_{PIDY} = 10.8[s]$$

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{dead} = 50 * 0.06$

$$T_{dead} = 3[s]$$

Estos datos arrojan que el tiempo de ejecución del sistema de control después del despegue del cuadricóptero fue de 10,8 s, en adelante como se puede observar en la Figura 110, el cuadricóptero se mantiene aproximadamente oscilando en el punto deseado.

Comportamiento de la posición del eje x con respecto al tiempo

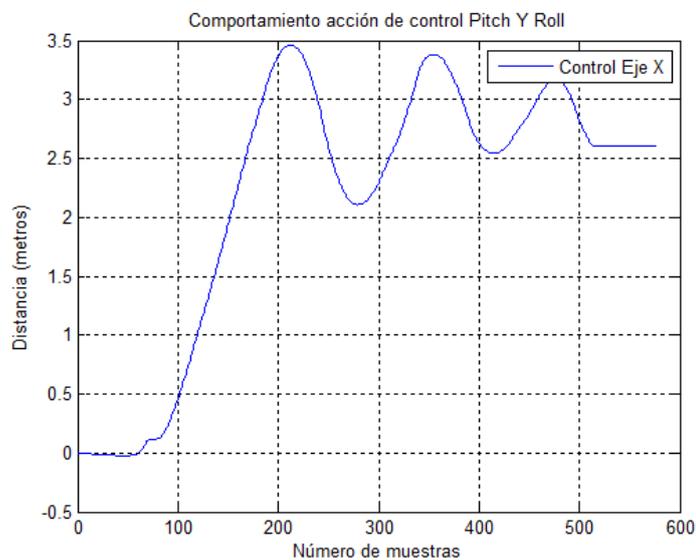


Figura 110. Respuesta en el tiempo del eje x, POS2

Número de muestras $\rightarrow m_{POS2} = 577$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS2 $\rightarrow t_{POS2} = m_{POS2} * t_s \Rightarrow 577 * 0.06$

$$t_{POS2} = 34.62[s]$$

CAPÍTULO 6

Tiempo de estabilización eje X $\rightarrow T_{PIDX} = 120 * 0.06$

$$T_{PIDX} = 7.20[s]$$

Tiempo muerto \rightarrow es el tiempo en que el cuadricóptero inicializa sus componentes y la acción de control de posición $\rightarrow T_{dead} = 50 * 0.06$

$$T_{dead} = 3[s]$$

Tabla 21. Respuesta del sistema de control POS2

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	10.8	15
Eje Y	7.20	15

Elaborado por: Autores del proyecto

Experimento POS3

Características de POS3

Tabla 22. Características del experimento POS3

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	0
Eje Y	0	-3

Elaborado por: Autores del proyecto

El cuadricóptero tiende una desviación negativa a lo largo del semiplano negativo, sin embargo a la ejecución del sistema de control retoma su trayecto hacia el punto de destino que se encuentra a lo largo del eje `y` negativo, con la condicionante que en el eje `x` su valor ideal es 0, como se aprecia en la Figura 111 este valor tiene a oscilar dependiendo de su control aplicado hacia el eje `x`.

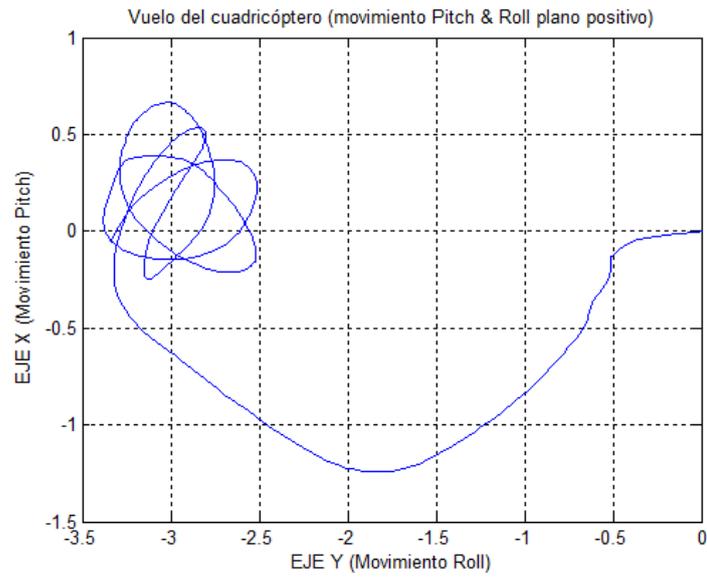


Figura 111. Desplazamiento en Matlab del cuadricóptero desde el punto inicial hacia el punto final del experimento POS2.



Fotografía 16. Vuelo real del cuadricóptero desde el punto inicial al punto final experimento POS3.

CAPÍTULO 6

Comportamiento de la posición del eje y con respecto al tiempo

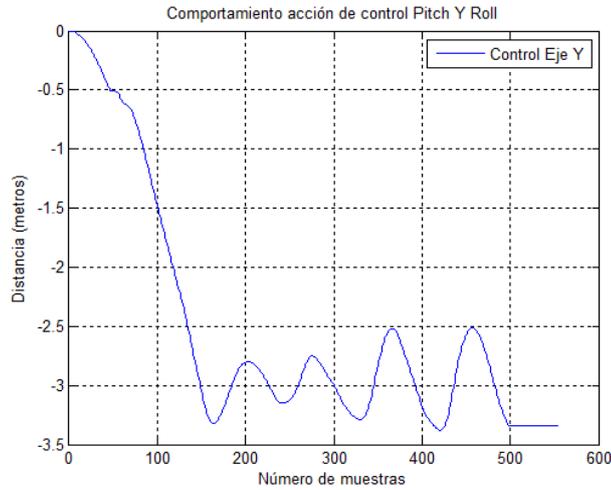


Figura 112. Respuesta en el tiempo del eje y, POS3

Número de muestras $\rightarrow m_{POS3} = 555$

Tiempo de muestreo $\rightarrow ts = 0.06[s]$

Tiempo de experimentación – POS3 $\rightarrow t_{POS3} = m_{POS3} * ts \Rightarrow 555.06$

$$t_{POS2} = 33.30[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 200 * 0.06$

$$T_{PIDY} = 12[s]$$

Tiempo muerto \rightarrow No hay tiempo muerto registrado por la gráfica 112

Comportamiento de la posición del eje x con respecto al tiempo

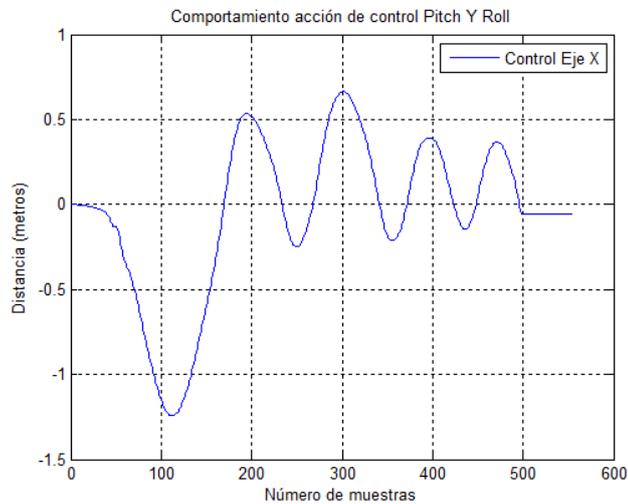


Figura 113. Respuesta en el tiempo del eje x, POS3

CAPÍTULO 6

Número de muestras $\rightarrow m_{POS3} = 555$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS3 $\rightarrow t_{POS3} = m_{POS3} * t_s \Rightarrow 555 * 0.06$

$$t_{POS2} = 33.30[s]$$

Tiempo de estabilización eje X $\rightarrow T_{PIDX} = 120 * 0.06$

$$T_{PIDX} = 7.20[s]$$

En este punto es importante destacar que no se puede calcular el tiempo de acción de control de posición ya que al ser un valor nulo, el sistema de control para este eje `x` deja de actuar, sin embargo debido al movimiento de vuelo del cuadricóptero que se aprecia en la Figura 113, existen fluctuaciones de valores entre los límites desde $\{-1; 1\}$, esto corresponde a la exactitud y al margen de error que se considera admisible. En adelante esto se presentará cuando una de las coordenadas del punto final de los ejes sea nula.

Tabla 23. Respuesta del sistema de control POS3

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	0	19
Eje Y	12	17

Elaborado por: Autores del proyecto

Experimento POS4

Características de POS4

Tabla 24. Características del experimento POS4

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	-3
Eje Y	0	-3

Elaborado por: Autores del proyecto

CAPÍTULO 6

En la Figura 114, el AR.Drone 2.0 se dirige hacia el plano negativo de las coordenadas del experimento POS4. Se aprecia que existe un pico en la respuesta del control ya que el cuadricóptero desde el punto inicial tiene un pequeño desplazamiento hacia los ejes positivos, pero una vez que entra en marcha el control retoma su trayectoria hacia el punto final.

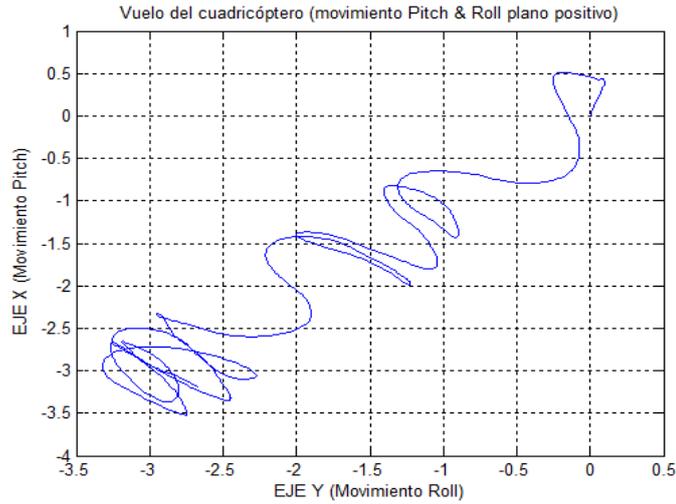


Figura 114. Desplazamiento en Matlab del cuadricóptero desde el punto inicial hacia el punto final del experimento POS4.



Fotografía 17. Vuelo real del cuadricóptero desde el punto inicial al punto final experimento POS4

CAPÍTULO 6

Comportamiento de la posición del eje y con respecto al tiempo

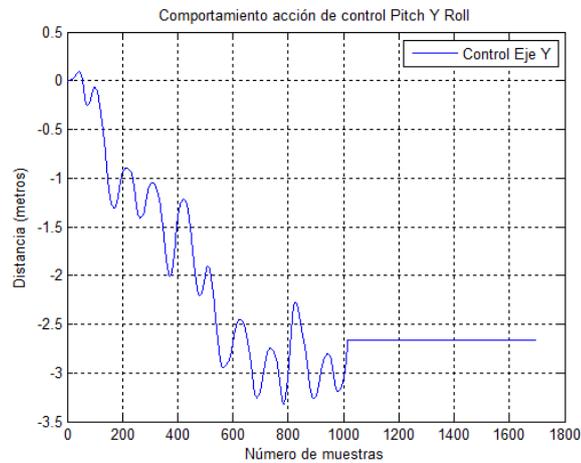


Figura 115. Respuesta en el tiempo del eje y, POS4

Número de muestras $\rightarrow m_{POS4} = 1696$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS4 $\rightarrow t_{POS4} = m_{POS4} * t_s \Rightarrow 1696 * 0.06$

$$t_{POS2} = 101.76[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 550 * 0.06$

$$T_{PIDY} = 33[s]$$

Comportamiento de la posición del eje x con respecto al tiempo

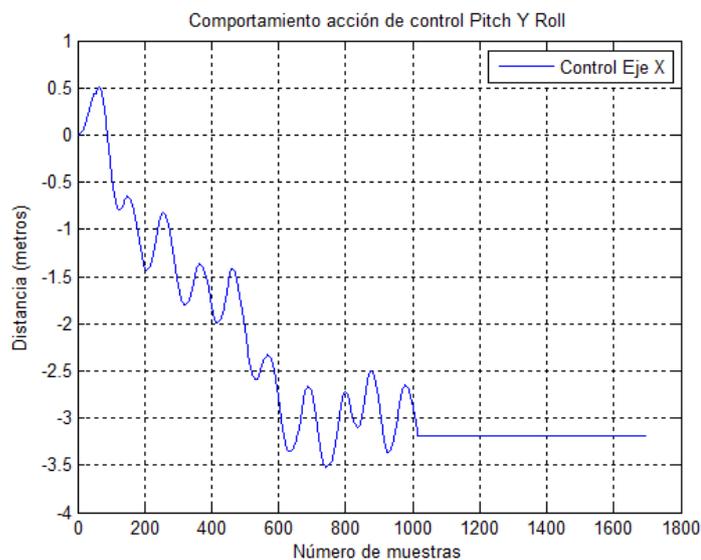


Figura 116. Respuesta en el tiempo del eje x, POS4

CAPÍTULO 6

Número de muestras $\rightarrow m_{POS4} = 1696$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS4 $\rightarrow t_{POS4} = m_{POS4} * t_s \Rightarrow 1696 * 0.06$

$$t_{POS2} = 101.76[s]$$

Tiempo de estabilización eje X $\rightarrow T_{PIDX} = 600 * 0.06$

$$T_{PIDX} = 36[s]$$

Tabla 25. Respuesta del sistema de control POS4

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	33	15
Eje Y	33	16

Elaborado por: Autores del proyecto

Experimento POS5

Características de POS5

Tabla 26. Características del experimento POS5

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	-2
Eje Y	0	0

Elaborado por: Autores del proyecto

En la Figura 117, el AR.Drone 2.0 se dirige hacia el eje `x` negativo. El pico existente en la acción de control produce un aumento de trayectoria haciendo que el cuadricóptero se desplace más allá de la coordenada establecida, en el transcurso del tiempo logra rectificar y comienza a trasladarse al punto final.

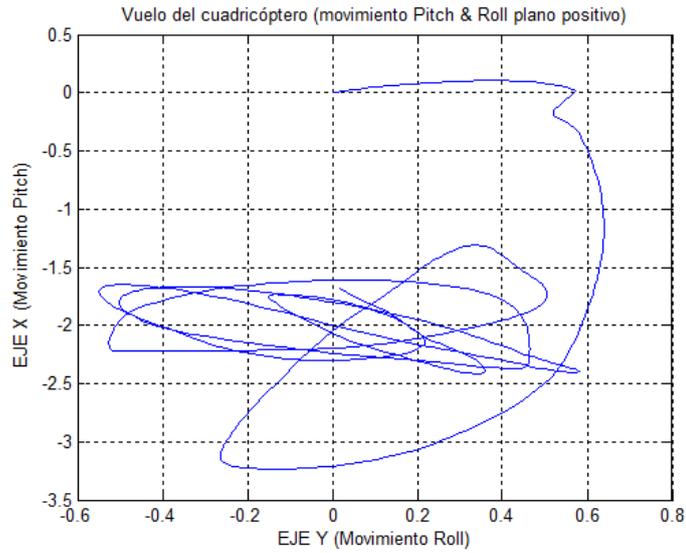


Figura 117. Desplazamiento en Matlab del cuadricóptero desde el punto inicial hacia el punto final del experimento POS5



Fotografía 18. Vuelo real del cuadricóptero desde el punto inicial al punto final experimento POS5

CAPÍTULO 6

Comportamiento de la posición del eje `y` con respecto al tiempo

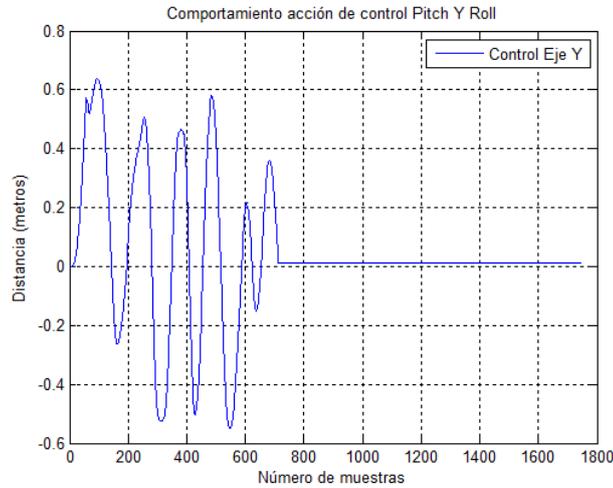


Figura 118. Respuesta en el tiempo del eje `y`, POS5

Número de muestras $\rightarrow m_{POS5} = 1747$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS5 $\rightarrow t_{POS5} = m_{POS5} * t_s \Rightarrow 1747 * 0.06$

$$t_{POS5} = 104.82[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 700 * 0.06$

$$T_{PIDY} = 42[s]$$

Cabe destacar que en el instante 42s, el control es más efectivo ya que se anula todas las oscilaciones que presenta el vuelo en tiempo real del cuadricóptero.

Comportamiento de la posición del eje `x` con respecto al tiempo

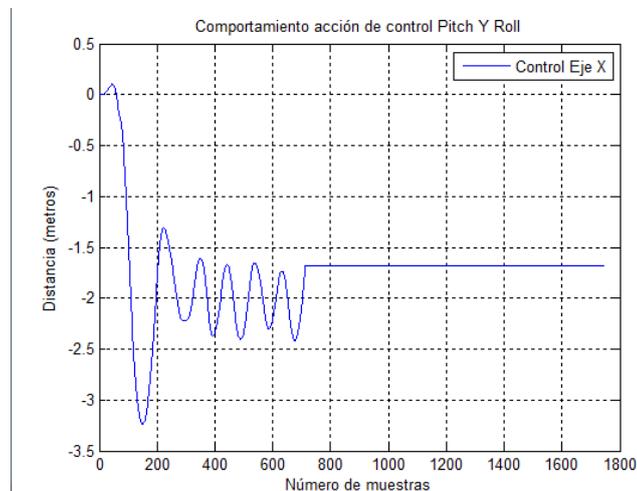


Figura 119. Respuesta en el tiempo del eje `x` POS5

CAPÍTULO 6

Número de muestras $\rightarrow m_{POS5} = 1747$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS5 $\rightarrow t_{POS5} = m_{POS5} * t_s \Rightarrow 1747 * 0.06$

$$t_{POS5} = 104.82[s]$$

Tiempo de estabilización eje X $\rightarrow T_{PIDX} = 100 * 0.06$

$$T_{PIDX} = 6[s]$$

Tabla 27. Respuesta del sistema de control POS5

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	6	20
Eje Y	42	16

Elaborado por: Autores del proyecto

Experimento POS6

Características de POS6

Tabla 28. Características del experimento POS6

Coordenada	Punto Inicial [m]	Punto Final [m]
Eje X	0	0
Eje Y	0	3

Elaborado por: Autores del proyecto

En la Figura 120, el AR.Drone 2.0 se dirige hacia el eje `y` positivo. El pico de la respuesta de la acción de control establece movimiento hacia el plano negativo en un tiempo razonable pero no despreciable, de igual forma como en los anteriores experimentos el control PID logra rectificar este error y regula la trayectoria logrando que el cuadricóptero llegue al punto final.

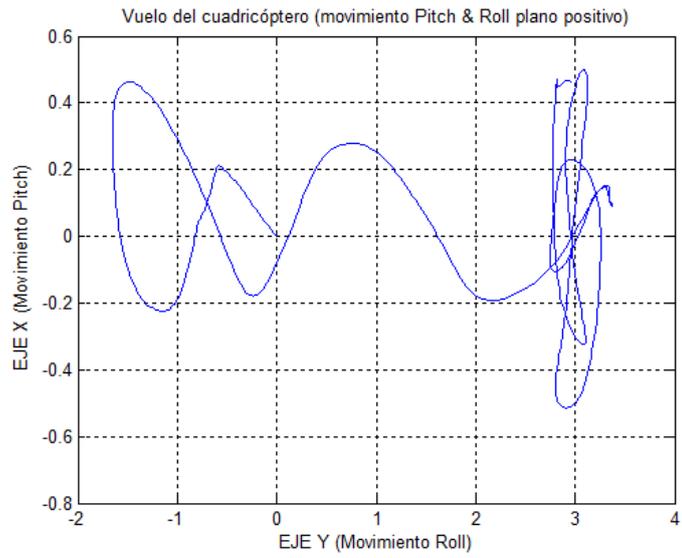


Figura 120. Desplazamiento en Matlab del cuadricóptero desde el punto inicial hacia el punto final del experimento POS6



Fotografía 19. Vuelo real del cuadricóptero desde el punto inicial al punto final experimento POS6

CAPÍTULO 6

Comportamiento de la posición del eje y con respecto al tiempo

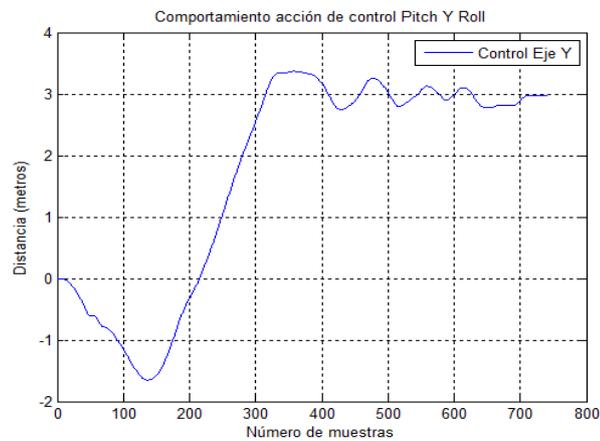


Figura 121. Respuesta en el tiempo del eje 'y', POS6

Número de muestras $\rightarrow m_{POS6} = 742$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS6 $\rightarrow t_{POS6} = m_{POS6} * t_s \Rightarrow 742 * 0.06$

$$t_{POS6} = 44.52[s]$$

Tiempo de estabilización eje Y $\rightarrow T_{PIDY} = 310 * 0.06$

$$T_{PIDY} = 18.60[s]$$

Comportamiento de la posición del eje x con respecto al tiempo

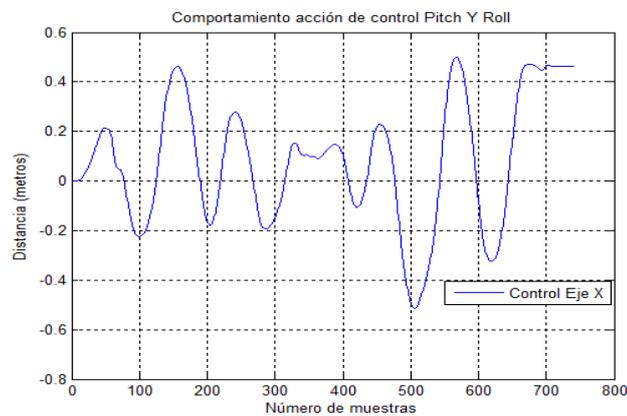


Figura 122. Respuesta en el tiempo del eje 'x', POS6

CAPÍTULO 6

Número de muestras $\rightarrow m_{POS6} = 742$

Tiempo de muestreo $\rightarrow t_s = 0.06[s]$

Tiempo de experimentación – POS6 $\rightarrow t_{POS6} = m_{POS6} * t_s \Rightarrow 742 * 0.06$

$$t_{POS6} = 44.52[s]$$

Tiempo de estabilización eje X $\rightarrow T_{PIDX} = 100 * 0.06$

$$T_{PIDX} = 6[s]$$

El control es aplicable desde la muestra inicial, el punto requerido es 0 por tanto el cuadricóptero logra estar al margen del objetivo deseado.

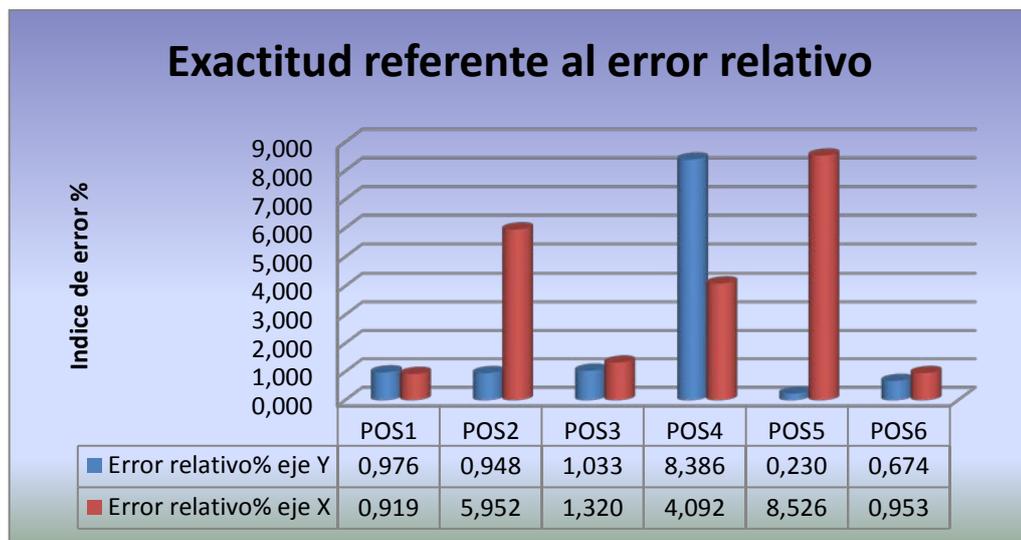
Tabla 29. Respuesta del sistema de control POS6

Coordenadas	Tiempo de Establecimiento [s]	Sobrepico [%]
Eje X	0	15
Eje Y	18.6	15

Elaborado por: Autores del proyecto

Exactitud y Precisión

Para sustentar la fiabilidad del sistema de control diseñado, se analiza la exactitud al llegar el cuadricóptero al punto final deseado. Analizado las pruebas de campo en el punto anterior, la Gráfica 3 muestra el porcentaje de error en cada uno de los 6 experimentos realizados.

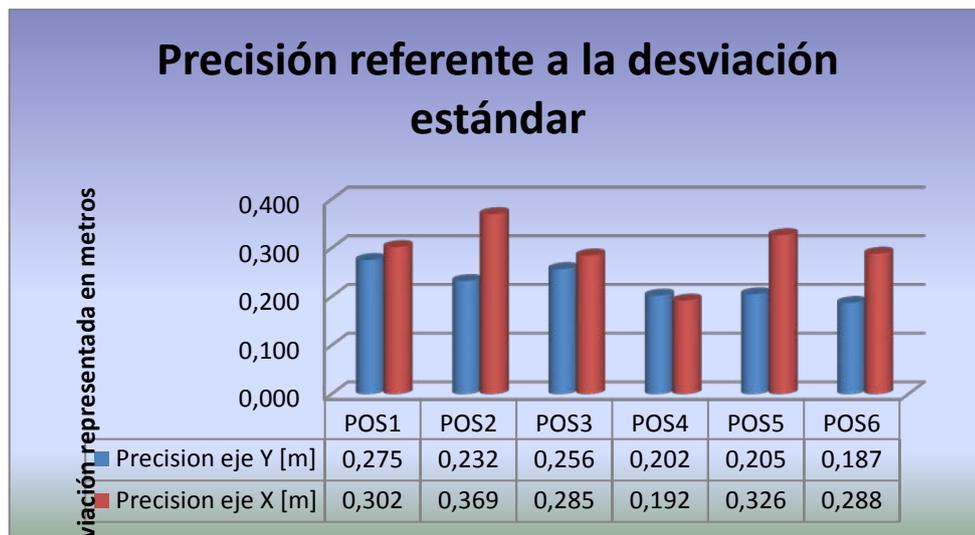


Gráfica 3. Exactitud de la posición en cada uno de los experimentos realizados

CAPÍTULO 6

El error más grande es alrededor del 8% en el experimento POS4 y el experimento POS5, un error que se establece cuando el cuadricóptero tiende a desplazarse al punto final deseado en coordenadas negativas, sin embargo este dato relevante no compromete en absoluto el buen desempeño del sistema de control.

La precisión se calcula con la desviación estándar, este parámetro describe hasta que distancia expresada en metros el cuadricóptero tiende a desviarse una vez que llega al punto final deseado, como se aprecia en la Gráfica 4, la desviación más grande entre todos los experimentos se refiere a POS2, desviándose casi 30cm desde la posición de punto final deseado.



Gráfica 4. Precisión de la posición en cada uno de los experimentos realizados

Otro factor interesante que presenta la Gráfica 4, es la precisión en el eje `y` es mejor en comparación con la precisión en el eje `x`.

Comparando la exactitud y precisión de los primeros experimentos (EXP1, EXP2) con los subsiguientes (POS1, POS2, POS3, POS4, POS5, POS6) y que se aplicó el ajuste del sistema de control en el hardware y ajuste fino de los parámetros del AR.Drone 2.0, se deduce que el enfoque a reemplazar el sistema electrónico y de hélices ayudo a que el porcentaje de error disminuya de manera considerable.

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

- Se diseña e implementa un control de posición para el cuadricóptero AR.Drone 2.0 de PARROT. Empleando técnicas de identificación paramétrica para el modelo y un control PID con el método de Ziegles-Nichols para determinar sus parámetros, en el cual se presenta las pruebas y resultados del sistema real efectuados en el Capítulo 6.
- El Software TrackDrone lite 2.0 fue de gran ayuda en la consecución del proyecto, ya que puede manejar la comunicación WIFI con el cuadricóptero, y permitió implementar el modelo y el algoritmo de control de posición diseñado.
- El rango de los ángulos Roll y Pitch para la acción de movimiento del cuadricóptero son entre $[-1; 1]$, con este antecedente los experimentos realizados en el transcurso del desarrollo del proyecto se logró notar variaciones de velocidad de vuelo con respecto a los valores que se ejecutó, se concluye que con un ángulo de entrada de 0.5 y su correspondiente valor negativo -0.5, el AR.Drone 2.0 presenta un vuelo normal con una velocidad apreciable, y permiten capturar los datos de posición a través del software TrackDrone lite 2.0.
- El sensor ultrasonido presenta problemas al momento de despegar en superficies tales como tierra, grava, césped, y existe una medición errónea de parte de este sensor, se

CAPÍTULO 7

- concluye que al despegar desde una superficie de material de cemento o cerámica la acción de elevación no presenta ningún problema.
- Se debe tomar como precedente que los datos generados por los experimentos de vuelo difieren del ambiente en donde se realiza, existe una gran diferencia con los datos tomados por las pruebas de campo realizadas en altitudes nominales a 500 metros sobre el nivel del mar, con respecto a los datos realizadas en alturas que sobrepasan los 1500 metros sobre el nivel del mar. Esta discusión plantea un problema en la realización del modelado del sistema, por tal motivo el resultado final en base al control diseñado y programado para el cuadricóptero será incompatible en estas dos alturas.
- Se concluye que a una altura de 500 metros sobre el nivel del mar, el AR.Drone 2.0 presenta estabilidad, y un comportamiento lineal e invariante con el tiempo, comparando con alturas que sobrepasan los 1000 metros sobre el nivel del mar.
- En los experimentos realizados existen ruidos y tiempos muertos, para identificar y modelar se acondiciona los datos eliminándolos.
- Utilizando la herramienta de identificación de Matlab denominado “ident”, resulta sencillo obtener los distintos modelos, se concluye que los resultados obtenidos permiten afirmar que dichos modelos se aproximan al comportamiento real del cuadricóptero en un 59.04% para la posición `x` en tiempo continuo y 83.73% para la posición `y` en tiempo continuo.

CAPÍTULO 7

- Es posible utilizar la técnica de desacople para cada eje de movimiento haciendo dos sistemas de simple entrada y simple salida, facilitando el problema y la dificultad, diseñando un control clásico y un controlador PID.
- El diseño de control empleando es el método de Ziegles-Nichols para ajustar los parámetros y la estrategia es desacoplar las posiciones (x, y), en lazo cerrado que permitió controlar la posición del cuadricóptero, inclusive ante la presencia de perturbaciones.
- Con las especificaciones dinámicas iniciales del control PDI diseñado pudieron ser mejorados los parámetros mediante un ajuste fino, mejorando el tiempo de establecimiento de 16.5s a 2.87s, siendo la sobre oscilación 9.51% a 13.4% para la posición `x`, y de 16.7s a 1.28s, siendo la sobre oscilación 7.16% a 0%, para la posición `y`.
- Después de haber cubierto todas las etapas de diseño e implementación del control de posición, se realiza pruebas del sistema real, analizando los parámetros del controlador. En el análisis de resultados se observa errores de posición menores al 8%, y precisiones menores a 50cm.

7.2 Recomendaciones

- Al realizar pruebas de campo con el cuadricóptero, es necesario utilizar un computador que cuente con un procesador I3 o posteriores, y una memoria RAM de 4GB, acorde a las nuevas generaciones de procesadores en la actualidad. Esto se debe al procesamiento de cálculo en tiempo real que debe ejecutar cuando esta enlazado con el vehículo en el instante de vuelo.
- Al ser un modelo de caja negra, es decir que depende de los datos experimentales que arroja el sistema, estos datos dependen del lugar geográfico donde estas pruebas de campo se realice, como sugerencia primordial el modelado del sistema se debe realizar en un mismo ambiente con las mismas condiciones climáticas para que no exista incoherencias en el diseño del control.
- Se recomienda al usuario realizar los vuelos destinados para el modelamiento y posterior visualización del control en una superficie considerablemente grande, con la intención de salvaguardar las partes mecánicas del cuadricóptero específicamente los motores, parte importante que al sufrir algún golpe estos motores son propensos a averiarse con facilidad.
- Si el usuario tiene un conocimiento de reparación electrónica se puede solucionar de forma inmediata los motores que han sido averiados por golpes, tomando en cuenta la microelectrónica que se presenta en la tarjeta de control de los motores. Se puede reponer los elementos dañados asociados a esta tarjeta de control por otros de similar características, pero como una solución temporal más no permanente.

CAPÍTULO 7

- Las hélices del cuadricóptero disminuyen en su durabilidad y rendimiento debido a las constantes pruebas en el modelamiento y que se le sometió en todo el progreso de la investigación, de tal manera que es necesario cambiar de manera circunstancial para no tener ninguna lectura errónea en el instante de crear una estructura de control de posición.
- La desventaja de la batería del AR.Drone 2.0 es que dura tan solo 12 minutos durante el vuelo, y se necesitan 90 minutos para su recarga por lo que se recomienda tener una o más baterías extras.

BIBLIOGRAFÍA

Behar, M. (2003). *Identificación y Control Adaptativo*. Madrid, España: Editora: Isabel Capella, Técnico editorial: Marta Caicoya.

Android. (05 de 04 de 2013). *Android*. Recuperado el 01 de 06 de 2013, de <http://www.android.com/>

Apple. (02 de 02 de 2013). *Ipod/Iphone/Ipad*. Recuperado el 01 de 06 de 2013, de <http://www.apple.com/es/ipad/>

AR.Drone Control App for Windows and Microsoft .Net. (Abril de 2013). Recuperado el Martes de Febrero de 2014, de <http://dronehacks.com/2010/12/10/ar-drone-control-app-for-windows-and-microsoft-net/>

Carvalho, S. D., & Prado, M. L. (s/f). *Controle Adaptativo Auto-ajustável de um Processo Térmico*. Feira de Santana - Bahia, Brasil: Departamento de Tecnologia, Universidade Estadual de Feira de Santana.

Rodriguez Ramirez, C. B. (2005). *Analisis de sistemas de espacios de estados Identificacion de sistemas Control Adaptativo Control Predictivo*. Apuntes de Ingeniería de control. Universitatis Litteraria e Hispalensis. México.

Enesbilgen. (s/f). *Math Works Research Internship Project*. Universidad de Boston.

Guillén, E. L. (2002). *Identificación de sistemas. Aplicación al modelado de un motor de continua*. Madrid, España: Universidad de Alcalá de Henares.

Inta. (06 de 05 de 2006). *Instituto Nacional de Técnica Aeroespacial*. Recuperado el 05 de 06 de 2013, de http://www.inta.es/doc/programasaltatecnologia/avionesnotripulados/siva_web.pdf

Linux. (01 de 04 de 2013). *Linux*. Recuperado el 01 de 06 de 2013, de <http://www.linux.org/>

Maluf, N., & Williams, K. (2004). *Introduction to Microelectromechanical Systems Engineering*. Norwood.

Mathworks. (2013). *Mathworks*. Recuperado el 07 de 05 de 2013, de <http://www.mathworks.com/>

Microsoft. (20 de 05 de 2013). *Microsoft*. Recuperado el 01 de 06 de 2013, de <http://www.microsoft.com/en-us/default.aspx>

Microsoft®. (2011). *Visual Studio Team Foundation Server® 2010, Service Pack 1*. Recuperado el 12 de 12 de 2014, de <http://www.microsoft.com/en-us/download/details.aspx?id=20506>

Ogata, K. (1998). *Ingeniería de control moderna*. Mexico. Pearson Prentice Hall: Tercera edición.

Ortiz T, H. (1998). *Instrumentación y sistemas de control*. Sangolquí. Editorial ESPE: Primera edición.

Parrot. (2010). *AR-Drone 2.0 Parrot*. Recuperado el 02 de 05 de 2013, de <http://ardrone2.parrot.com/>

Piskorski, s., Brulez, N., Eline, P., & D`Haeyer, F. May(May 17, 2011). *AR.Drone Developer Guide*. SDK 1.7. Francia. Parrot: Segunda edición.

Stephane Piskorski, N. B. (February 24, 2011). *AR.Drone Developer Guide*. SDK 1.6. Francia. Parrot: Primera edición.

El Telegrafo. (24 de 04 de 2013). www.letelegrafo.com.ec. Recuperado el 15 de 05 de 2013, de El telegrafo: <http://www.telegrafo.com.ec/noticias/guayaquil/item/invasiones-son-controladas-con-vuelos-no-tripulados.html>

UPV. (2012). *CPOH control y optimización*. Recuperado el 28 de 01 de 2014, de <http://cpoh.upv.es/en/research/software/item/236-trackdrone-lite.html>

York, T. M. (1998). *Tesla Society* . Recuperado el 05 de 06 de 2013, de <http://www.teslasociety.com/robotics.htm>

GLOSARIO

- **Cuadricóptero**

Vehículo volador en forma de cruz, que acopla 4 hélices en el final de cada una de sus extremidades, gracias a esta estructura versátil el movimiento del cuadricóptero es más estable y dinámico en contraparte con el helicóptero.

- **Polipropileno**

De estructura termoplástica es un material que sirve para la fabricación de las carcasas internas, externas del cuadricóptero y de su estructura modular, este material es utilizado por tener menor peso específico, tiene propiedades dieléctricas y evita el traspaso de la humedad, ventaja importante para salvaguardar los componentes electrónicos importantes para la acción del movimiento.

- **Ultrasónico**

Fenómeno que aplica una emisión de sonido de alta frecuencia hacia una determinada superficie, esta emisión se refleja y vuelve hacia el dispositivo que lo emite. Utilizado en sensores para medir distancias o altitudes tal como utiliza el cuadricóptero para su medición de altitud.

- **Magnetómetro**

Dispositivo empleado por el cuadricóptero para establecer referencias de orientación con respecto al usuario y el terminal que se utiliza, ya sea un dispositivo táctil o un computador.

- **Giroscopio**

Utilizado para monitorear y mantener la orientación del cuadricóptero en el espacio, la dinámica del giroscopio establece un punto de referencia en el eje z, por lo tanto

cuando el cuadricóptero tiene un movimiento irregular que afecte las hélices el giroscopio monitorea este cambio de posición y cierra la comunicación con el usuario con el fin de salvaguardar las piezas mecánicas del vehículo.

- **Brushless**

Definición dada a los motores sin escobillas que emplea el cuadricóptero, son utilizados gracias a que no produce energía calorífica producida por el rozamiento en el instante de cambio de polaridad del rotor. Ventaja importante ya que la temática de los motores del AR.Drone 2.0 es cambiar constantemente de la polaridad y de esta forma presentar la acción de despegue, vuelo y aterrizaje.

- **Busybox**

Programa que encierra varias aplicaciones más pequeñas con la finalidad de ejecutar software para el control manual del cuadricóptero, comúnmente busybox es una herramienta que se ejecuta en sistemas operativos Linux, sin embargo también aplica a sistemas Android.

- **Navdata**

Paquete especial del software de programación del AR.Drone 2.0, este paquete entrega tanto al usuario como la aplicación HMI todos los parámetros técnicos de vuelo y navegación del cuadricóptero.

ACTA DE ENTREGA

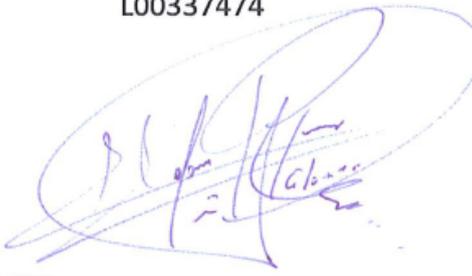
El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE, desde:

Sangolquí, 30 de octubre de 2015

ELABORADO POR:

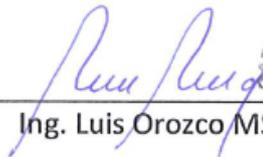


Sr. Germmy Franklin Coloma Vallejo
1718867987
L00337474



Sr. Kid Coleman Oyos Piarpuesan
1716769334
L00332110

AUTORIDAD



Ing. Luis Orozco MSc.



DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL