

**DISEÑO E IMPLEMENTACIÓN DE UNA TARJETA DE ADQUISICIÓN DE DATOS  
INALÁMBRICA PARA EL CONTROL DEL HELICÓPTERO 2-DOF DE QUANSER®.**

ANGELICA MARIA HERNANDEZ BONILLA  
FERNANDO NICOLAS VARGAS PLAZAS

**TRABAJO DE GRADO PARA OPTAR POR EL TITULO DE  
INGENIERO ELECTRONICO**

**DIRECTOR**  
ANDRES ALBERTO LADINO LOPEZ I.E. M.Sc

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
DEPARTAMENTO DE ELECTRÓNICA  
BOGOTÁ DC, 2013

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERIA  
CARRERA DE INGENIERIA ELECTRONICA**

RECTOR MAGNÍFICO: PADRE JOAQUÍN EMILIO SÁNCHEZ GARCÍA S.J

DECANO ACADÉMICO: ING. JORGE LUIS SANCHEZ TELLEZ M.Sc

DECANO DEL MEDIO UNIVERSITARIO: PADRE SERGIO BERNAL RESTREPO S.J

DIRECTOR DE CARRERA: ING. JAIRO ALBERTO HURTADO LONDOÑO Ph.D

DIRECTOR DE PROYECTO: ING. ANDRES ALBERTO LADINO LOPEZ M.Sc

## **ARTÍCULO 23 DE LA RESOLUCIÓN No 13 DE JUNIO DE 1946**

“La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Sólo velará porque no se duplique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia”.

## TABLA DE CONTENIDO

<b>1. INTRODUCCION.....</b>	<b>7</b>
<b>2. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS .....</b>	<b>8</b>
2.1 OBJETIVO GENERAL .....	8
2.2 OBJETIVO ESPECIFICOS .....	8
<b>3. ESPECIFICACIONES .....</b>	<b>8</b>
3.1 ESPECIFICACIONES DE HARDWARE .....	8
3.2 ESPECIFICACIONES DE SOFTWARE .....	8
3.3 ENTRADAS Y SALIDAS DEL SISTEMA.....	9
<b>4. MARCO TEÓRICO.....</b>	<b>9</b>
4.1 NETWORKED CONTROL SYSTEM.....	9
4.2 WIRELESS NETWORKED CONTROL SYSTEM .....	10
4.3 ADQUISICIÓN DE DATOS .....	10
4.4 PROTOCOLO ZIGBEE Y MÓDULOS XBEE .....	12
4.4.1 PROTOCOLO ZIGBEE .....	12
4.4.2 MÓDULOS XBEE.....	12
4.5 QUARC®.....	13
<b>5. DIAGRAMA EN BLOQUES Y DESCRIPCION GENERAL.....</b>	<b>14</b>
<b>6. DESARROLLO TEORICO .....</b>	<b>16</b>
6.1 HELICOPTERO 2-DOF.....	16
6.1.1 MODELO MATEMÁTICO .....	16
6.1.2 COMPONENTES DEL HELICOPTERO .....	18
6.1.2.1 MOTORES DE YAW Y PITCH: .....	18
6.1.2.2 HELICES.....	18
6.1.2.3 ENCODERS .....	18
6.2 TARJETA DE ADQUISICION .....	19
6.2.1 MICROCONTROLADOR .....	19
6.2.1.1 DESARROLLO EN EL MICROCONTROLADOR .....	20
6.2.1.1.1 COMUNICACIÓN SERIAL VIA UART .....	21
6.2.1.1.2 ALGORITMO DE CAPTURA, MULTIPLEXACION Y ENVIO DE DATOS DEL HELICOPTERO.....	22
6.2.1.1.3 ALGORITMO DE RECEPCION Y DEMULTIPLEXACION DE LOS DATOS PROVENIENTES DE QUARC® .....	23
6.2.1.1.4 SALIDA DE DATOS POR MEDIO DE SEÑALES PWM.....	24
6.2.2 OSCILADOR .....	25
6.2.3 MÓDULOS XBEE/XBEE - PRO OEM RF.....	26
6.2.3.1 PINES QUE SOPORTAN MÚLTIPLES FUNCIONES .....	27
6.2.3.2 TARJETA XBEE EXPLORER USB.....	28
6.2.3.3 CONFIGURACION DEL MODULO XBEE EN LA TARJETA DE ADQUISICION ..	29
6.2.4 FILTRO PASA BAJOS ANALOGO .....	30
6.2.5 ACONDICIONAMIENTO DE NIVEL.....	32
6.2.6 ACONDICIONAMIENTO DE SEÑAL.....	32
6.3 MODULOS DE POTENCIA (UPM) .....	33
6.4 ADQUISICION Y PROCESAMIENTO DE DATOS EN MATLAB® .....	34
6.4.1 BLOQUE STREAM SERVER DE QUARC®.....	34
<b>7. CIRCUITO IMPRESO.....</b>	<b>37</b>
<b>8. ARQUITECTURA PARA LA CODIFICACIÓN DE DATOS EN SIMULINK.....</b>	<b>38</b>
<b>9. PROTOCOLO DE PRUEBAS.....</b>	<b>41</b>
9.1 PRUEBAS EN TARJETA DE ADQUISICION.....	41
9.2 PRUEBAS DE RECEPCION DE DATOS EN SIMULINK .....	45
9.3 PRUEBAS SOBRE EL SISTEMA DE CONTEO DE PULSOS DE LOS ENCODERS .....	46
9.4 PRUEBAS DEL SISTEMA EN LAZO ABIERTO.....	48

9.5	PRUEBAS DEL SISTEMA EN LAZO CERRADO TARJETA INALAMBRICA-Q8 USB.....	50
9.6	PRUEBAS DEL SISTEMA EN LAZO CERRADO TARJETA INALAMBRICA .....	55
<b>10.</b>	<b>ANÁLISIS DE RESULTADOS.....</b>	<b>57</b>
10.1	ERROR EN LOS VALORES DE SALIDA DEL CIRCUITO .....	57
10.2	DISTANCIA ENTRE LOS MÓDULOS XBEE AL EJECUTAR EL CONTROL .....	59
10.3	RENDIMIENTO DEL SISTEMA DE CONTEO DE LOS ENCODERS .....	59
10.4	ANÁLISIS DE PRUEBAS OBTENIDAS DE LAZO ABIERTO .....	60
10.5	ANÁLISIS DE PRUEBAS OBTENIDAS DE LAZO CERRADO .....	62
10.5.1	ANÁLISIS DE RESULTADOS EN LA COMUNICACIÓN.....	62
10.5.2	COMPARACION CON MODELO TEORICO .....	63
10.5.3	ANÁLISIS DEL EFECTO EN EL TIEMPO DE MUESTREO: .....	65
<b>11.</b>	<b>ESTUDIO DE COSTOS .....</b>	<b>66</b>
<b>12.</b>	<b>TRABAJOS E INVESTIGACIONES A SEGUIR .....</b>	<b>66</b>
<b>13.</b>	<b>CONCLUSIONES.....</b>	<b>67</b>
<b>14.</b>	<b>BIBLIOGRAFIA Y FUENTES DE INFORMACION .....</b>	<b>68</b>

## ÍNDICE DE FIGURAS

Figura 1.	Estructura típica de un NCS [3] .....	9
Figura 2.	Estructura general de un sistema de adquisición de datos .....	10
Figura 3.	Trasmisor de Datos [8].....	11
Figura 4.	Receptor de Datos [8] .....	11
Figura 5.	Tecnologías en el rango de 2.4 GHz [9].....	12
Figura 6.	Logo de QuaRC® .....	13
Figura 7.	Diagrama en bloques del sistema de adquisición .....	14
Figura 8.	Helicóptero 2-DOF .....	16
Figura 9.	Diagrama de cuerpo libre del helicóptero.....	16
Figura 10.	Motores de yaw y pitch. ....	18
Figura 11.	Hélices de yaw y pitch. ....	18
Figura 12.	Pines de encoder .....	19
Figura 13.	Señales de encoder en cuadratura.....	19
Figura 14.	Microcontrolador PIC16F876A .....	20
Figura 15.	Comunicación serial UART asíncrona. ....	21
Figura 16.	Detección de pulsos por modo 4X .....	22
Figura 17.	Máquina de estados, conteo de pulsos de encoders .....	22
Figura 18.	Diagrama de flujo de algoritmo de recepción de datos.....	24
Figura 19.	Oscilador de cristal conectado a microcontrolador.....	26
Figura 20.	Módulo XBEE .....	26
Figura 21.	Tarjeta XBee Explorer USB.....	28
Figura 22.	Circuito integrado FT232RL.....	28
Figura 23.	Señal típica de PWM.....	31
Figura 24.	Espectro de la señal de PWM.....	31
Figura 25.	Filtro pasabajos de tercer orden.....	32
Figura 26.	Esquemático del restador para bajar el nivel de voltaje DC de las señales PWM.....	32
Figura 27.	Configuración para amplificador no inversor .....	32
Figura 28.	Conexiones típicas en UPM .....	33
Figura 29.	Conexión de cables para amplificación en UPM.....	33
Figura 30.	Bloque Stream Server de QuaRC® para Simulink [22] .....	34
Figura 31.	Capa inferior del circuito impreso implementado. ....	37
Figura 32.	Capa superior del circuito impreso implementado. ....	38
Figura 33.	Bloques de codificación de datos de encoders .....	40
Figura 34.	Diagrama de flujo del algoritmo en MATLAB® .....	40
Figura 35.	Conversión entre los datos de ciclo útil a su voltaje correspondiente .....	42

Figura 36. Bloques en Simulink que permite enviar los valores de voltaje para generar las señales de PWM.....	42
Figura 37. Salida de señal PWM en el microcontrolador al enviar el valor 0 V desde Simulink.....	43
Figura 38. Salida de señal PWM en el microcontrolador al enviar el valor 2 V desde Simulink.....	43
Figura 39. Salida de señal PWM en el microcontrolador al enviar el valor -3 V desde Simulink.....	43
Figura 40. Señales de PWM en el microcontrolador, Pitch en -4.5 V (amarillo) y Yaw en 2 V (azul).....	44
Figura 41. Señal de PWM en el microcontrolador al enviar el valor 3 V desde Simulink.....	44
Figura 42. Señal de PWM a la salida del circuito restador que adecua la señal entre $\pm 1.65$ V.....	44
Figura 43. Señal de control hacia la UPM de 3 V a la salida del filtro pasa bajo con ganancia de 2.9. ....	45
Figura 44. Salida de recepción del bloque stream server de QuaRC® mostrando los conteos de los encoders.....	46
Figura 45. Señal de Pitch concatenada.....	46
Figura 46. Señal de Yaw concatenada.....	46
Figura 47. Sistema de simulación y control de Quanser® para helicóptero 2-DOF.....	48
Figura 48. Respuesta de pitch lazo abierto con 2 V de amplitud.....	49
Figura 49. Respuesta de yaw lazo abierto con 2 V de amplitud.....	49
Figura 50. Respuesta de pitch lazo abierto con 3 V de amplitud.....	49
Figura 51. Respuesta de yaw lazo abierto con 3 V de amplitud.....	49
Figura 52. Respuesta de pitch lazo cerrado LQR y LQI con señal de referencia de 10 grados.....	52
Figura 53. Respuesta de yaw lazo cerrado LQR y LQI con señal de referencia de 10 grados.....	53
Figura 54. Respuesta de pitch lazo cerrado con control proporcional discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados.....	53
Figura 55. Respuesta de yaw lazo cerrado con control proporcional discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados.....	53
Figura 56. Respuesta de pitch lazo cerrado con LQR y LQI con señal de referencia de 10 grados.....	54
Figura 57. Respuesta de yaw lazo cerrado con LQR y LQI con señal de referencia de 10 grados.....	54
Figura 58. Respuesta de pitch lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados.....	54
Figura 59. Respuesta de yaw lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados.....	55
Figura 60. Respuesta de pitch lazo cerrado con controladores LQR y LQI con señal de referencia de 10 grados.....	56
Figura 61. Respuesta de pitch lazo cerrado con controladores LQR y LQI con señal de referencia de -10 grados.....	56
Figura 62. Respuesta de yaw lazo cerrado con controladores LQR y LQI con señal de referencia de 10 grados.....	56
Figura 63. Respuesta de pitch lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados.....	56
Figura 64. Respuesta de pitch lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de -10 grados.....	57
Figura 65. Respuesta de yaw lazo cerrado con controladores discreto 2 y proporcional continuo con señal de referencia de 10 grados.....	57
Figura 66. Desviación de los datos medidos con respecto a los valores esperados en la salida del circuito.....	58
Figura 67. Grafica al comparar los valores medidos y su diferencia con respecto a los valores esperados.....	58
Figura 68. Rendimiento del sistema para conteo de los encoders para 3 diferentes velocidades.....	60
Figura 69. Respuesta de pitch lazo abierto con Q8-USB y tarjeta inalámbrica.....	61
Figura 70. Respuesta de yaw lazo abierto con Q8-USB y tarjeta inalámbrica con ganancia de -2.....	61
Figura 71. Respuesta de yaw lazo abierto con Q8-USB y tarjeta inalámbrica con ganancia de 3.....	61
Figura 72. Respuesta de pitch en lazo cerrado con controlador LQR a una velocidad de 115200 baudios.....	62
Figura 73. Diagrama en bloques del modelo teórico al ser discretizado con un zoh.....	63
Figura 74. Respuesta en lazo cerrado teórico con control LQR y control LQI.....	63
Figura 78. Respuesta de pitch en lazo cerrado con controlador LQR con Q8-USB con retenedor de orden cero con a) 0.001 s, b) 0.032 s.....	65

## ÍNDICE DE TABLAS

Tabla 1. Entradas y salidas del sistema .....	9
Tabla 2. Parámetros del helicóptero 2-DOF .....	17
Tabla 3. Pines correspondientes al módulo XBee .....	26
Tabla 4. Conexión de los pines para el Circuito Integrado FT232RL.....	28
Tabla 5. Comparación de características entre XBee serie 1 y serie 2 .....	29
Tabla 6. Configuración de los módulos XBee.....	30
Tabla 7. Pestaña principal de los parámetros configurables para el bloque Stream Server de QuaRC®. ...	35
Tabla 8. Pestaña secundaria de los parámetros configurables para el bloque Stream Server de QuaRC®.	36
Tabla 9. Parámetros configurables para el URI del protocolo serial.....	36
Tabla 10. Comparación de valores .....	45
Tabla 11. Pruebas del rendimiento del sistema de conteo de los Encoders a diferentes velocidades .....	47
Tabla 12. Desempeño de los controladores con una velocidad de transmisión de 38.400 Baudios. ....	51
Tabla 13. Desempeño de los controladores con una velocidad de transmisión de 57.600 Baudios. ....	51
Tabla 14. Desempeño de los controladores con una velocidad de transmisión de 115.200 Baudios. ....	52
Tabla 15. Error en la salida hacia la UPM de la tarjeta de adquisición de datos.....	57
Tabla 16. Resultados de las pruebas de distancia. ....	59
Tabla 17. Características dinámicas y estáticas en lazo cerrado conexión Tarjeta Diseñada-Q8 USB .....	64
Tabla 18. Características dinámicas y estáticas en lazo cerrado de la Tarjeta Diseñada .....	64
Tabla 19. Características dinámicas y estáticas en lazo cerrado del Modelo Teórico .....	64
Tabla 20. Costos reales de los recursos utilizados para la realización del Trabajo de Grado.....	66

## ÍNDICE DE SIGLAS Y ABREVIATURAS

NCS: Networked Control System
DAQ: Sistema de adquisición de datos
USB: Universal serial bus
ADC: Analog to digital converter
DAC: Digital to analog converter
DOF: Degrees of freedom
PID: Proporcional integral derivativo
UART: Universal Asynchronous Receiver and Transmitter
API: Application Programming Interface
PWM: Pulse width modulation
UPM: Universal power module
LQR: Linear quadratic regulator
[V]: Voltio
[A]: Amperio
[Ω]: Ohmio
[g]: Gramo
[m]: Metro
[Hz]: Hertz
[rad]: radián
[s]: Segundo
[N]: Newton
[F]: Faradio
[baud]: Baudio
[bps]: Bit por segundo

## 1. INTRODUCCION

Uno de los retos de los últimos años en tecnología ha sido la integración de diferentes tipos de redes, tanto cableadas como inalámbricas, que permitan la interacción de dispositivos de control y monitoreo. Por tanto, la implementación de un sistema de control para la regulación remota de una planta resulta oportuna y favorable para su uso en un ambiente de laboratorio.

La implementación de un sistema de control (actuadores) se realiza típicamente conectando éstos elementos mediante cables para la transmisión de paquetes de datos. Esta interacción entre controlador y planta no tiene en cuenta otras tecnologías como las inalámbricas que en ocasiones son más eficientes y traen mayores beneficios tanto económicos como de implementación.

Las conexiones inalámbricas pueden traer grandes ventajas tanto de tipo económico como de tipo técnico. Económicamente, una conexión cableada acarrea grandes gastos por la longitud de los cables, el tipo de cableado para una determinada velocidad (cable UTP, fibra óptica, etc.), los costos de instalación y los gastos en mantenimiento a largo de la vida útil de la misma, mientras que una conexión inalámbrica solo requiere de un montaje inicial y una configuración. Los costos de tecnologías de este tipo son menores frente a conexiones alambreadas. Cualquiera de estas tecnologías de conexión inalámbrica resulta ser eficiente para realizar una adecuada implementación [1].

Por otro lado, éste tipo de implementación trae grandes beneficios en múltiples aplicaciones, puesto que la conectividad es mucho más sencilla y permite tener mayor versatilidad en el manejo físico de la planta, facilidad para realizar la configuración cableada y modificaciones sin cambiar la posición de los elementos a pesar de la distancia. Otro punto a favor de estas tecnologías es que satisfacen velocidades de transmisión suficientes para la aplicación que se va a trabajar [9].

El trabajo de grado a realizar cambia la perspectiva tradicional de conexión de sistemas de control y se enfoca en estabilizar el lazo a través de una interconexión entre el controlador y la planta de forma inalámbrica utilizando el estándar de comunicación inalámbrica IEEE 802.15.4. El sistema consta de una tarjeta de adquisición de datos conectada por medio de cables a la planta de Quanser® (Helicóptero 2-DOF) y a las UPM correspondientes; la tarjeta tendrá embebido un módulo XBee que se conectará de forma inalámbrica a un controlador que será implementado desde MATLAB® con uno de los algoritmos clásicos de control para lograr la estabilización de la planta ante éstas condiciones de conexión.

La tarjeta de adquisición se basa en la lectura y codificación de los datos que envía el helicóptero 2DOF de sus ángulos de cabeceo y guiñada (pitch y yaw) [2], para luego enviarlos de forma inalámbrica por medio de un módulo XBee hacia otro módulo XBee que se encuentra conectado al computador vía USB y por medio de MATLAB®, se realiza la interacción planta-controlador. Posteriormente, se envían las señales de control resultantes del algoritmo hacia la tarjeta de adquisición a través de los módulos XBee, y ésta los envía de forma analógica por las UPM hacia los motores que realizan el control de la posición de los ángulos del helicóptero.

El documento se estructura en la especificación de los requerimientos básicos y las operaciones que deben ser realizadas por el sistema de interconexión inalámbrica para el correcto control del helicóptero, la presentación de los desarrollos y trabajos realizados en éste campo, teniendo en cuenta las ventajas y desventajas que han tenido, la descripción general del sistema propuesto, así como el desarrollo teórico y detallado de cada una de las partes que componen el diseño de la tarjeta de adquisición y de la interconexión inalámbrica. Por último se presenta una propuesta de estimación de costos del proyecto y de su implementación.



## **2. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS**

### **2.1 OBJETIVO GENERAL**

Diseñar e implementar un sistema para la interacción inalámbrica entre una ley de control y la plataforma helicóptero de 2 grados de libertad Quanser® 2D Heli.

### **2.2 OBJETIVO ESPECIFICOS**

- Implementar una conexión inalámbrica para el helicóptero de dos grados de libertad que permita una comunicación bidireccional entre el sistema de control y la plataforma.
- Probar y evaluar una ley de control que permita la estabilización del sistema bajo condiciones ideales del canal.
- Integrar el sistema a la plataforma QuaRC® que permite la implementación de controladores (compensadores, control por realimentación de estados) mediante una interconexión inalámbrica.

## **3. ESPECIFICACIONES**

El sistema para la interacción inalámbrica entre un controlador diseñado en MATLAB® y el helicóptero 2DOF de Quanser® cumple con los siguientes requerimientos:

### **3.1 ESPECIFICACIONES DE HARDWARE**

La tarjeta de adquisición cuenta con lo siguiente:

- Niveles de voltaje entre -5 V y 5V.
- Dos entradas digitales para lectura de encoders ópticos con resolución de 14 bits.
- Dos entradas análogas para manejo opcional de otros módulos de Quanser®.
- Interfaz con módulo Zigbee para comunicación inalámbrica.
- Etapa de acondicionamiento de señal para envío de señales a los módulos de potencia de Quanser®.
- Dos salidas análogas.
- Velocidad de transmisión y recepción de 38400 baudios.

### **3.2 ESPECIFICACIONES DE SOFTWARE**

- MATLAB® 7.9 (R2009b) con Simulink o superior.
- Módulo de QuaRC® en MATLAB® con bloques de comunicación.
- Windows XP, Vista ó Windows 7.

El software de implementación en MATLAB® necesita que el computador donde se vaya a ejecutar cumpla con los siguientes requerimientos:

- Procesador Intel Pentium, Celeron, AMD con velocidad de 1 GHz o superior.
- Disco duro de 10 GB o superior.
- Memoria RAM con 1 GB o superior.
- Puerto USB para conexión de módulo Zigbee

### 3.3 ENTRADAS Y SALIDAS DEL SISTEMA

Las entradas y salidas del sistema se realizan teniendo en cuenta como sistema principal la tarjeta de adquisición de datos.

Tabla 1. Entradas y salidas del sistema

SEÑAL	TIPO DE SEÑAL	UNIDADES
Señal de encoder de pitch	ENTRADA	Conteos por revolución
Señal de encoder de yaw	ENTRADA	Conteos por revolución
Señal analógica de control de pitch amplificada	SALIDA	Voltios (V)
Señal analógica de control de yaw amplificada	SALIDA	Voltios (V)

El diagrama de bloques detallado explicando la interconexión del sistema y sus señales correspondientes se encuentra en la sección 5.

## 4. MARCO TEÓRICO

### 4.1 NETWORKED CONTROL SYSTEM

Los sistemas de control en red (NCS) han sido una de las principales líneas de investigación en las que se centra el mundo académico y la industria y se ha convertido en un área multidisciplinar. Con esta línea de investigación en crecimiento, es importante consolidar los conocimientos más recientes y la información para mantenerse al día con las necesidades de investigación [3].

Un NCS es un sistema de control en el que los lazos de control están cerrados a través de una red en tiempo real. La característica principal de un NCS es que el control y las señales de realimentación se intercambian entre los componentes del sistema en forma de paquetes de información a través de una red.

La funcionalidad de un NCS típico se establece mediante el uso de cuatro elementos básicos:

- **Sensores:** Para adquirir información,
- **Controladores:** Para proporcionar decisiones y órdenes,
- **Actuadores:** Para llevar a cabo los comandos de control.
- **Red de comunicación:** Para permitir el intercambio de información.

En la figura 1 se muestra una estructura típica de un sistema de control en red.

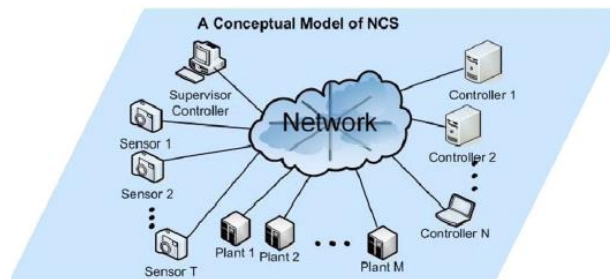


Figura 1. Estructura típica de un NCS [3]

La característica más importante de un NCS es que conecta el ciberespacio con el espacio físico, por lo tanto permite la ejecución de varias tareas de larga distancia. Además, los NCS eliminan el cableado

innecesario, que reduce la complejidad y el costo global en el diseño y la aplicación de los sistemas de control. También pueden ser fácilmente modificados o mejorados si se adicionan sensores, actuadores y controladores a ellos con un costo relativamente bajo y sin grandes cambios en su estructura. Por otra parte, con el intercambio eficiente de datos entre los controladores, los NCS son capaces de fusionar fácilmente información mundial para tomar decisiones inteligentes sobre grandes espacios físicos [3].

Sus posibles aplicaciones son numerosas y abarcan una amplia gama de industrias, tales como: la exploración terrestre y espacial, el acceso en entornos peligrosos, automatización de fábricas, diagnóstico remoto y solución de problemas, instalaciones experimentales, robots domésticos, aviones, automóviles, monitoreo de las plantas de manufacturación y tele-operación.

## 4.2 WIRELESS NETWORKED CONTROL SYSTEM

Los sistemas de control en red (NCS) se están aplicando a través de redes inalámbricas debido a la necesidad de movilidad de los nodos en muchas aplicaciones. Estos sistemas son conocidos como Sistemas de control en red inalámbricos (WNCS). Los más simples WNCS incluyen una planta y un controlador de punto a punto con la comunicación entre ellos de forma remota o inalámbrica [4].

Se han desarrollado por la necesidad de las operaciones móviles, instalaciones flexibles, y despliegue rápido para muchas aplicaciones. Cuando los sistemas inalámbricos se incluyen a los NCS, la fiabilidad y el tiempo son más difíciles de satisfacer, debido a las características nocivas de los canales de radio tales como la interferencia electromagnética de otras redes que se encuentren en la misma frecuencia, la trayectoria de la señal al ser enviada por un medio no confinado y la susceptibilidad a pérdidas de comunicación cuando existen interferencias físicas. La mayoría asume que en los sistemas de control los datos recogidos son exactos, oportunos y sin pérdidas. Esos son los desafíos que los WNCS deben alcanzar [5].

La diferencia entre WNCS y NCS es que para el primero el control remoto es por microprocesadores, y la información se puede enviar fiable a través de las redes digitales compartidas, incluso conexiones inalámbricas. En los sistemas de distribución en una red inalámbrica, sensores diferentes pueden tener varias localizaciones y sus mediciones deben ser codificadas y enviadas una por una a través de los diferentes enlaces de las redes inalámbricas.

## 4.3 ADQUISICIÓN DE DATOS

La adquisición de datos es el proceso de muestreo de señales que miden las condiciones físicas del mundo real y las convierten en muestras digitales resultantes en valores numéricos que pueden ser manipulados por un computador. Los sistemas de adquisición de datos (abreviado con las siglas DAS o DAQ) suelen convertir señales analógicas en valores digitales para su procesamiento. Los componentes de los sistemas de adquisición de datos incluyen:

- Transductor que convierten los parámetros físicos en señales eléctricas.
- Circuitos de acondicionamiento de señal para convertir las señales de los sensores en señales que se puedan convertir a valores digitales.
- Conversores Análogo/Digital (tratamiento), que convierten las señales del sensor acondicionado en valores digitales.

En la figura 2 se muestra un diagrama de bloques general de un sistema de adquisición de datos y a continuación se hace una breve explicación de los bloques

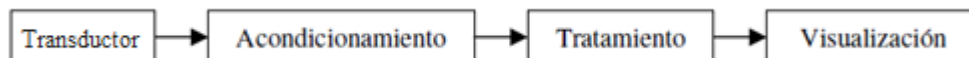


Figura 2. Estructura general de un sistema de adquisición de datos

El primer paso en la adquisición de datos es detectar y convertir las características físicas tales como: la presión, temperatura, velocidad de flujo, y la posición en una señal eléctrica. Esto se hace usando un transductor. Éste es un dispositivo que convierte una cantidad física en una señal eléctrica y posee propiedades importantes como la linealidad. Una relación lineal entre el nivel de excitación y de salida de un transductor es deseable [6]. Sin embargo, muchas veces la salida de un transductor es no lineal. Si el comportamiento no lineal se puede cuantificar, entonces la salida se hace predecible. La amplitud de salida de un transductor es otra propiedad importante. Una mayor amplitud de salida es deseable debido a que proporciona mediciones más precisas con menor susceptibilidad al ruido.

El segundo paso consiste en el acondicionamiento de la señal. Un acondicionador de señal es generalmente necesario para amplificar, atenuar y/o filtrar la señal que es producida por el transductor. Típicamente, la señal producida por el transductor está en el rango de mili-voltios y el acondicionador de señal se amplifica a un intervalo predeterminado (valores utilizados normalmente son 0 a 5 voltios o 0 a 10 voltios).

Posteriormente, se realiza la conversión analógica a digital. El método de manipulación de datos del computador es estrictamente binario, es decir, todo es alto o bajo. Los fenómenos físicos en los que se basan los transductores para crear una señal analógica permanecen. Esto introduce la necesidad de conversión análoga a digital (A/D) por medio de conversores A/D (dispositivos) que interactúan entre los dos tipos de información. Los eventos digitales también se deben tener en cuenta para niveles de transistor a transistor (TTL lógicos de voltaje). Después de la conversión A/D, la información está lista para ser introducida en el computador donde se almacena y/o manipula usando un paquete de software.

Existen varios tipos de conversores análogo-digital, lo más usados son los siguientes:

- De aproximaciones sucesivas
- Flash
- Sigma-delta

Otros tipos de conversores análogo-digital son: escalera, seguimiento, rampa, doble rampa, tensión-frecuencia, etc. [7].

El sistema de adquisición de datos juega un papel importante en los experimentos de ingeniería. Un sistema típico de adquisición de datos y de uso común es la tarjeta de adquisición de datos, que puede ser insertada en el PC y hacer un sistema basado en PC de adquisición de datos. Tomando ventaja de la computación de gran alcance y capacidad de procesamiento del PC, este tipo de sistema de adquisición de datos es fácil de usar y excelente en el análisis de datos. Por lo tanto, es ampliamente adoptado en ingeniería de medición y prueba [8]. Sin embargo, este tipo de producto de adquisición de datos tiene la desventaja de ser cableado, lo que no es conveniente para sistemas que necesitan renovación continua ya que los precios de mantenimiento son muy elevados.

En la figura 3 y figura 4 que se muestran a continuación se puede observar un sistema de adquisición de datos con antenas receptoras y trasmisoras.

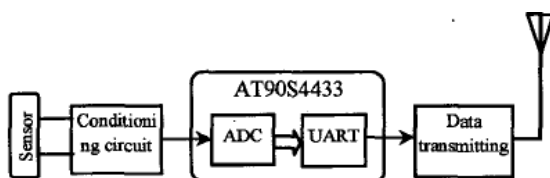


Figura 3. Trasmisor de Datos [8]

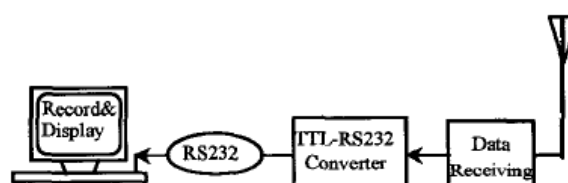


Figura 4. Receptor de Datos [8]

## 4.4 PROTOCOLO ZIGBEE Y MÓDULOS XBEE

### 4.4.1 PROTOCOLO ZIGBEE

ZigBee es una tecnología inalámbrica de corto alcance y bajo consumo originaria de la antigua alianza HomeRF y que se definió como una solución inalámbrica de baja capacidad para aplicaciones en el hogar como la seguridad y la automatización. Su objetivo no es obtener velocidades muy altas, ya que solo puede alcanzar una tasa de 20 a 250Kbps en un rango de 10 a 75 metros; sino obtener sensores cuyos transceptores tengan un muy bajo consumo energético. Lo anterior ha sido demostrado por dispositivos alimentados con dos pilas AA que pueden soportar varios años sin el cambio de baterías [9]. Por tanto, dichos dispositivos pasan la mayor parte del tiempo en un estado latente, es decir, durmiendo para consumir mucho menos. Opera en las bandas libres de 2.4Ghz, 858Mhz para Europa y 915Mhz para Estados Unidos.

En la figura 5 se puede ver el espectro de ocupación en las bandas del protocolo 802 (incluyendo ZigBee).

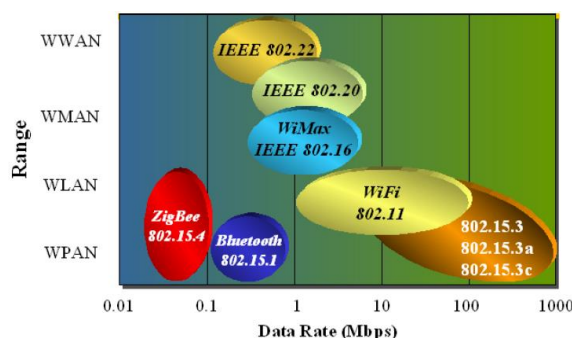


Figura 5. Tecnologías en el rango de 2.4 GHz [9]

En una red ZigBee puede haber hasta 254 nodos, sin embargo, según la agrupación que se haga, se pueden crear hasta 255 conjuntos/clusters de nodos con lo cual se puede llegar a tener 64770 nodos para lo que existe la posibilidad de utilizar varias topologías de red: en estrella, en malla o en grupos de árboles. Estas redes pueden abarcar una gran superficie por medio de un enrutamiento de saltos múltiples también conocido como multi-hop.

### 4.4.2 MÓDULOS XBEE

Existen dos versiones generales para los módulos XBee, los de la serie 1 y los de la serie 2 y XBee –Pro. Para el caso de este trabajo de grado se utilizó el módulo XBee de Digi International serie 1. A continuación se explica brevemente las principales diferencias entre estos módulos.

- **Serie 1:**  
Maneja el estándar 802.15.4 que se define:

**Capa física:** tipo de modulación y frecuencia (2.4 GHz, 16-QAM, 915 MHz y 868 MHz DSSS-BPSK) y el formato del paquete.

**Control de acceso al medio:** timing, asociación con un nodo coordinador, direccionamiento, prevención de colisiones, retransmisión, encriptación, ruteo de 2 saltos a través de un nodo coordinador, etc.

Su principal aplicación es para redes de topología Peer-to-Peer (Punto a Punto).

- **Serie 2 y XBee-Pro:**

Maneja el protocolo ZigBee, construido sobre el protocolo 802.15.4, y se define como:

**Capa de red:** Estilo AODV, que permite múltiples saltos utilizando el ruteo por redes mesh. Asociación a una red (no sólo a un nodo coordinador como el caso de 802.15.4). Más tipos de encriptación y autenticación.

**Capa de Aplicación:** Permite a los nodos exponer valores individuales, como sensores, switches o actuadores. Perfiles para varios tipos de dispositivos, etc.

En este caso el módulo XBee se usa generalmente en topologías de red Mesh que sean self-organizing.

Las diferentes versiones de módulos XBee no son compatibles unas con otras, es decir no se puede realizar una conexión entre un módulo XBee serie 1 con un módulo XBee-Pro.

#### 4.5 QUARC®

Es un software multifuncional que está integrado con la herramienta Simulink que posee MATLAB®. Se caracteriza por ser un ambiente de desarrollo multifuncional de control en tiempo real. Este software está diseñado para operar bajo los sistemas operativos de Windows.

Utilizando QuaRC® en Simulink permite implementar controladores que son convertidos en código de ejecución en tiempo real a través de tarjetas de adquisición o de otro tipo de conexiones (TCP/IP, UDP, serial, etc.), además algunos parámetros del control pueden ser modificados mientras el código se está ejecutando.

Otras características que posee son: ejecución de múltiples controladores al mismo tiempo, control distribuido, modelos de referencia y acceso a varios protocolos de comunicación, asimismo permite graficar los resultados del control en tiempo real y mandar los datos al espacio de trabajo de MATLAB®. [26]



Figura 6. Logo de QuaRC®

Otros parámetros a tener en cuenta son los siguientes:

- Compatibilidad con todos los bloques que posee Simulink, como matemáticos, continuos, discretos, osciloscopios, etc.
- Gráficas en tiempo real con la posibilidad de exportar los datos obtenidos a MATLAB®.
- Posibilidad de compilar código fuente de MATLAB® junto con el ambiente de Simulink.
- Soporta comunicación de tipo asíncrona con otros dispositivos que no son las tarjetas de Quanser®.
- Soporta protocolos de comunicación tal como TCP, UDP, ARCNET, etc.

## 5. DIAGRAMA EN BLOQUES Y DESCRIPCION GENERAL

El siguiente diagrama permite observar cada uno de los bloques que describen de forma detallada la adquisición de datos de la tarjeta a diseñar y la manera como se envían los datos de forma inalámbrica entre el helicóptero 2-DOF y el controlador diseñado en MATLAB®:

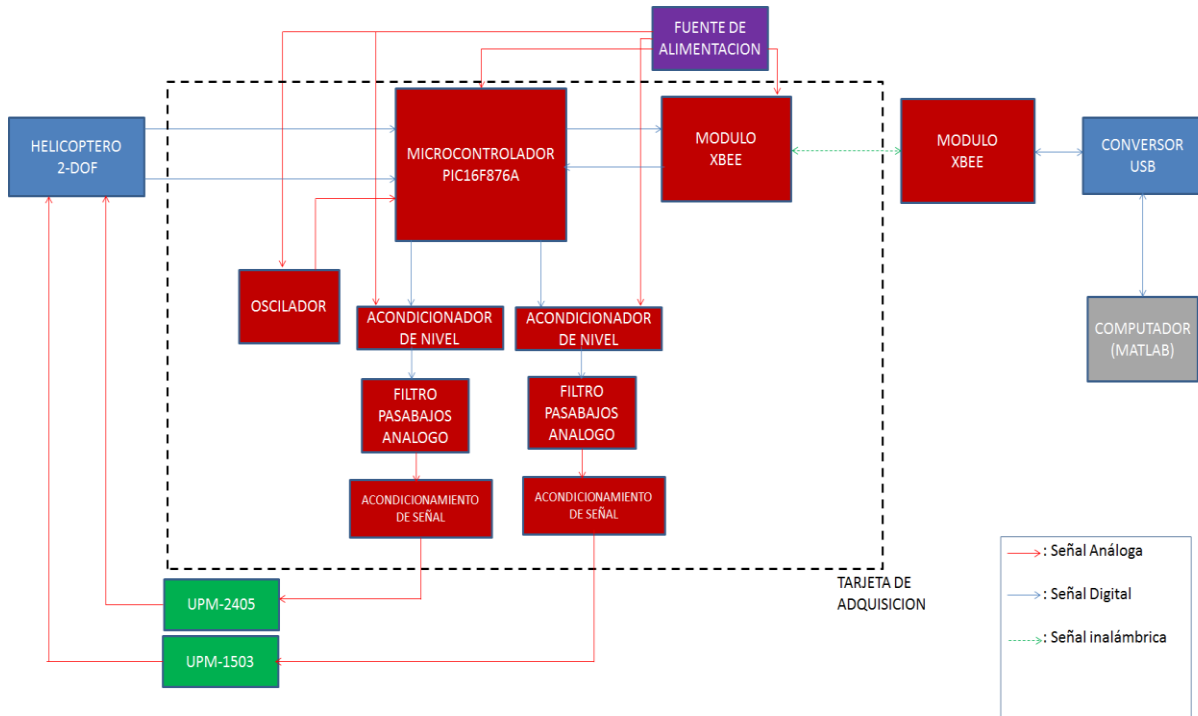


Figura 7. Diagrama en bloques del sistema de adquisición

La descripción general del diagrama es la siguiente, teniendo en cuenta que las líneas de color rojo corresponden a señales analógicas, las azules a señales digitales y la punteada a la señal inalámbrica:

- **HELICOPTERO 2-DOF:** El helicóptero envía las señales correspondientes a los ángulos de pitch y yaw por medio de dos encoders con un rango de posicionamiento de 4096 conteos/revolución y 8192 conteos/revolución respectivamente [10], a través de cables 5 PIN-stereo-DIN que se conectan a la tarjeta de adquisición con una polarización de 5V.
- **MICROCONTROLADOR:** Estas señales provenientes de los encoders de pitch y yaw son almacenadas y leídas por el microcontrolador PIC16F876A, que es capaz de leer varios encoders al tiempo y tiene una memoria de 8192 bytes [11], suficiente para la lectura y conteo de las señales de los encoders.
- **MODULO XBEE:** Luego de almacenar los datos de los encoders en el microcontrolador, éstos se envían a una entrada digital del módulo XBEE que se encuentra en la tarjeta de adquisición, por medio de una conexión serial UART que posee éste microcontrolador, de tal forma que por medio de la conexión NonBeacon que se explicará en la sección 6.2.3, se realice la comunicación inalámbrica entre este módulo y el que se conecta al computador vía USB. Los datos provenientes de los dos encoders son multiplexados en tiempo por medio del microcontrolador de tal forma que no haya peligro de colisión entre los paquetes enviados o en pérdida de información en la transmisión y que los datos puedan ser leídos de forma separada y correcta por MATLAB®.

- **COMUNICACIÓN EN MATLAB®:** Cuando el módulo que se conecta al computador recibe los datos provenientes de la tarjeta de adquisición, éstos son leídos por QuaRC®, a través de un bloque de comunicación cliente-servidor, lee éstos datos provenientes del helicóptero y los ejecuta de forma constante, teniendo así una actualización de las características del mismo y del lazo de control como tal.

El controlador que es implementado con una ley de control ya existente, es efectuado en un esquema digital y ejecutado en tiempo real gracias al módulo de QuaRC® que posee Simulink a la frecuencia de muestreo de la tarjeta de adquisición (1kHz) [12] y sus datos son enviados por el módulo USB hacia la tarjeta de adquisición. Dado que son enviados a la tarjeta los datos correspondientes a yaw y pitch, estos deben ser multiplexados en Simulink y cuando la tarjeta de adquisición los recibe, éstos son enviados por conexión serial UART hacia el microcontrolador, el cual se encarga de demultiplexarlos.

- **CONVERSION DIGITAL ANALOGA Y ACONDICIONAMIENTO DE SEÑAL:** debido a que la comunicación entre los módulos XBEE y el microcontrolador es digital y la señal de control que se dirige a los módulos de potencia UPM debe ser análoga, los datos provenientes de Simulink deben ser convertidos a señales análogas. Por consiguiente, las señales a la salida del microcontrolador (por medio de dos pines de salida) son señales de tipo PWM (Pulse width modulation) y son convertidas a señales analógicas DC mediante filtros pasabajos análogos y circuitos acondicionadores de señal cuyos aspectos de diseño se enunciarán más adelante.

Estas señales análogas correspondientes a las señales de control, son llevadas a cada una de las UPM de tal forma que sean amplificadas y de esta forma se entreguen al motor los niveles de voltaje adecuados para que sean manejados como se pretende con el control.

- **CONVERTIDOR USB:** La forma como se reciben los datos en el computador, que es directamente desde el módulo XBEE USB y procesando los datos del control en Simulink-QuaRC®, se escogió frente a otras opciones como la de diseñar un hardware de interconexión entre la Q4 y el módulo XBEE para realizar el cómputo en la Q4, puesto que el segundo implica el diseño de otra tarjeta que debe conectar de forma perfecta los dos elementos, lo que implica un diseño con especificaciones más exactas, como lo son inmunidad al ruido, componentes discretos con menores rangos de tolerancia, una programación extra del microcontrolador que realizará la lectura de datos de los dos elementos, etc. Además implica una elevación de costos y una utilización mayor de elementos en el sistema, lo que aumenta la complejidad del mismo y los riesgos de error son mucho mayores. Por otro lado, la programación e implementación en MATLAB®-Simulink implica la reducción de hardware a implementar y la reducción de costos del proyecto, además con la programación del software en ésta plataforma se asegura que la lectura y envío de datos lo hará con menos complicaciones que el otro método y la pérdida de información es mucho menor.

Respecto al diseño de la tarjeta de adquisición, se escogió la lectura y procesamiento de los datos de los encoders frente al diseño que realizan las tarjetas de Quanser® mediante registros, compuertas y contadores que son implementados de forma discreta, puesto que genera mayor facilidad en su implementación por ser sólo un componente y al implementar varios componentes de lectura y conteo de datos, se puede perder información por la inexactitud de los mismos o por fallas en la implementación de hardware como presencia de ruido en la tarjeta. Por otro lado, el microcontrolador realiza las operaciones básicas que se necesitan para éste proyecto y no requiere muchas otras que se encuentran en las tarjetas de Quanser®, que además generan una mayor complejidad en el diseño digital tanto de hardware como de software de la tarjeta lo que conlleva a una elevación de costos por la precisión que deben tener todos éstos componentes para la lectura eficiente de los datos.



## 6. DESARROLLO TEORICO

### 6.1 HELICOPTERO 2-DOF

El helicóptero 2-DOF de Quanser® es una plataforma experimental que consta de dos hélices controladas por motores DC. La hélice frontal se encarga de regular la elevación del cuerpo del helicóptero en sobre el eje denominado pitch. La hélice posterior regula el movimiento circular sobre el eje denominado yaw. Los ángulos pitch y yaw correspondientes a cada eje respectivamente, se miden mediante encoders de alta resolución. [8]



Figura 8. Helicóptero 2-DOF

#### 6.1.1 MODELO MATEMÁTICO

La descripción matemática del sistema dinámico del helicóptero para de del diagrama de cuerpo libre del mismo:

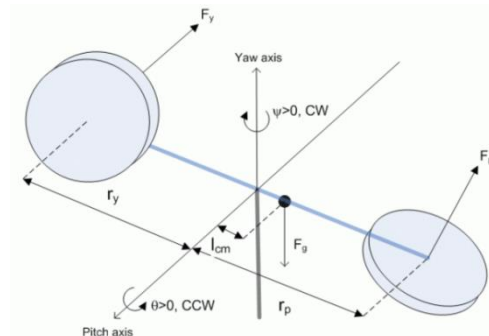


Figura 9. Diagrama de cuerpo libre del helicóptero

Este diagrama se basa en varias consideraciones:

La velocidad de rotación del helicóptero resulta de una fuerza  $F_p$  que actúa normal a éste a una distancia  $r_p$  del eje horizontal. Esta rotación causa un torque  $\tau_p$  en el eje del motor al igual que en el eje vertical ( $\tau_y$ ), donde también el motor genera una fuerza  $F_y$  a una distancia  $r_y$ .

Las ecuaciones de movimiento generadas por el diagrama de cuerpo libre son las siguientes [9]:

$$\ddot{\theta} = \frac{K_{pp}V_{m,p} + K_{py}V_{m,y} - m_{heli}gl_{cm} \cos(\theta) - B_{p.eq}\dot{\theta} - m_{heli}l_{cm}^2 \sin(\theta) \cos(\theta) \dot{\varphi}^2}{J_{eq,p} + m_{heli}l_{cm}^2}$$

$$\ddot{\varphi} = \frac{K_{yp}V_{m,p} + K_{yy}V_{m,y} - B_{y.eq}\dot{\varphi} + m_{heli}l_{cm}^2 \sin(\theta) \cos(\theta) \dot{\varphi}^2}{J_{eq,p} + m_{heli}l_{cm}^2}$$

Siendo,

Tabla 2. Parámetros del helicóptero 2-DOF

Abreviatura	Descripción	Valor
$J_{eq,p}$	Momento total de inercia sobre el eje horizontal (pitch)	0.0384 [kg.m <sup>2</sup> ]
$J_{eq,y}$	Momento total de inercia sobre el eje horizontal	0.0432 [kg.m <sup>2</sup> ]
$m_{heli}$	Masa total del helicóptero	1.3872 [kg]
$L_{cm}$	Longitud del centro de masa	0.186 [m]
$K_{pp}$	Constante de fuerza de empuje del motor de pitch	0.204 [N.m/V]
$K_{py}$	Constante de fuerza de empuje actuando en el eje pitch desde el motor de yaw	0.0068
$K_{yp}$	Constante de fuerza de empuje actuando en el eje yaw desde el motor de pitch	0.0219
$K_{yy}$	Constante de fuerza de empuje del motor yaw	0.072
$B_p$	Amortiguamiento viscoso equivalente sobre el eje pitch	0.800 [N/V]
$B_y$	Amortiguamiento viscoso equivalente sobre el eje yaw	0.318 [N/V]

Usando la fórmula de Euler-LaGrange, estas ecuaciones del sistema pueden ser derivadas y por tanto linealizadas alrededor del punto de equilibrio 0 en el espacio de estados, como se puede apreciar en el anexo 1:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

Siendo los vectores de estado, entrada y de salida los siguientes [8]:

$$x^T = [\theta(t) \ \varphi(t) \ \dot{\theta}(t) \ \dot{\varphi}(t)]$$

$$u^T = [V_{m,p} \ V_{m,y}]$$

$$y^T = [\theta(t) \ \varphi(t)]$$

Las matrices de estado son:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-B_p}{J_{eq,p} + m_{heli}l_{cm}^2} & 0 \\ 0 & 0 & 0 & \frac{-B_y}{J_{eq,y} + m_{heli}l_{cm}^2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{pp}}{J_{eq,p} + m_{heli}l_{cm}^2} & \frac{K_{py}}{J_{eq,p} + m_{heli}l_{cm}^2} \\ \frac{K_{yp}}{J_{eq,y} + m_{heli}l_{cm}^2} & \frac{K_{yy}}{J_{eq,y} + m_{heli}l_{cm}^2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

## 6.1.2 COMPONENTES DEL HELICOPTERO

El helicóptero 2DOF está compuesto principalmente por los siguientes componentes:

### 6.1.2.1 MOTORES DE YAW Y PITCH:

Son motores DC, siendo el motor de yaw el que actúa sobre la hélice trasera del helicóptero y el motor de pitch sobre la hélice delantera. El primero tiene una resistencia asociada de  $1.6 \Omega$  y una constante torque-corriente de  $0.0109 \text{ N.m/A}$  y el segundo tiene una resistencia de  $0.83 \Omega$  y una constante torque-corriente de  $0.0182 \text{ N.m/A}$ .

Las señales de control que llegan a los motores provenientes de la tarjeta de adquisición se conectan por medio de las fuentes de alimentación UPM2405 para el motor de pitch y la UPM 1503 para el motor de yaw, de tal modo que las señales de control sean amplificadas por éstos módulos antes de llegar a los motores. Los cables que se conectan de las UPM hacia los motores son del tipo 4 pin-stereo-DIN a 6 pin-stereo-DIN, uno de ganancia 5 que se conecta al motor de pitch y uno de ganancia 3 que se conecta al motor de yaw, como se detallará más adelante en la descripción de los módulos UPM. [10]



Figura 10. Motores de yaw y pitch.

### 6.1.2.2 HELICES

Las hélices de yaw y pitch están montadas sobre los ejes de los motores y sobre el blindaje de aluminio que las protege. La hélice y el motor de pitch tienen una constante de fuerza de empuje de  $0.104 \text{ N/V}$ , mientras que la hélice y el motor de yaw tienen una constante de fuerza de empuje de  $0.43 \text{ N/V}$ . [10]

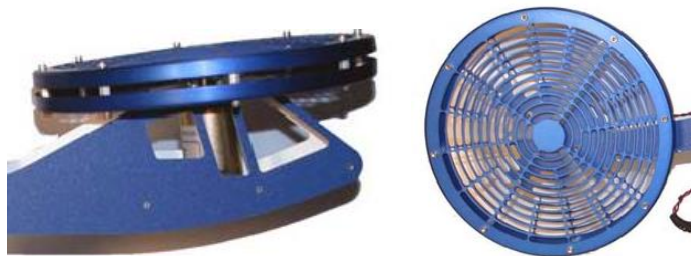


Figura 11. Hélices de yaw y pitch.

### 6.1.2.3 ENCODERS

El helicóptero tiene dos encoders digitales en cuadratura que miden los ángulos de pitch y yaw que tienen una resolución de 4096 conteos/revolución y 8192 conteos/revolución para pitch y yaw respectivamente. La posición efectiva de los encoders es de  $0.0879$  grados por revolución para pitch y  $0.0439$  grados por revolución para yaw [10]. Estos encoders tienen tres señales para un conteo bidireccional: canal A, canal B y canal I o pulso índice. Este último no se necesita para realizar el conteo pero es conveniente para calibración o como pulso de referencia para los otros canales. Por otro lado, tiene un canal para la alimentación del encoder que es de  $+5\text{V}$  y otro canal para referencia de tierra. [12]

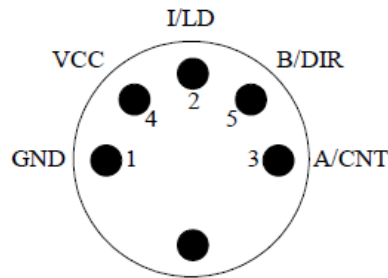


Figura 12. Pines de encoder

El funcionamiento del encoder en cuadratura se basa en dos sensores ópticos que generan pulsos digitales desfasados  $90^\circ$  uno del otro o en cuadratura, en éste caso A y B. Mediante ellas se pueden obtener datos de posición, dirección y velocidad de rotación de los ejes de los motores pitch y yaw. Suele usarse una convención en la que cuando la señal de A está por delante de B, el eje está rotando en sentido horario y al contrario, en sentido antihorario. La conexión entre los encoders y la tarjeta de adquisición se realiza mediante cables 5 pin stereo-DIN a 5 pin stereo-DIN de ganancia 1.

La señal índice puede estar sincronizada con A o B y es programada en éste caso para cargar el contador en el microcontrolador que toma los pulsos de cada encoder o para detenerlo dependiendo de los flancos de subida o de bajada que se detecten en cada momento [13]. En el caso del helicóptero 2-DOF, ésta señal viene deshabilitada desde la conexión inicial que tiene éste, por lo que no es necesario realizar el control de la señal en la tarjeta de adquisición.

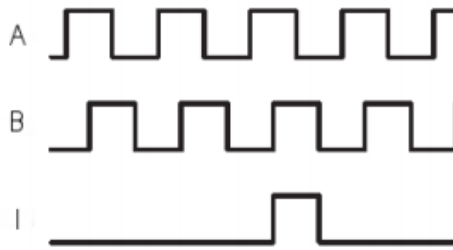


Figura 13. Señales de encoder en cuadratura

## 6.2 TARJETA DE ADQUISICION

En el diagrama de bloques se evidencia que la tarjeta de adquisición de datos a diseñar, cuenta con 6 bloques principales: Microcontrolador, módulo XBee, oscilador, filtro pasabajos análogo, acondicionamiento de señal y acondicionamiento de nivel. A continuación se hará una descripción detallada de cada uno de éstos.

### 6.2.1 MICROCONTROLADOR

Los datos provenientes de los ángulos de yaw y pitch por medio de los dos encoders deben ser leídos y guardados constantemente de tal manera que el módulo XBee pueda ser capaz de enviar ésta información seguidamente al módulo que se encuentra en el computador. Para realizar ésta acción, se decide utilizar un microcontrolador que sea capaz de recibir las señales de dos encoders al mismo tiempo y que además posea una memoria lo suficientemente grande para guardar los conteos por cada revolución que tienen estos encoders [8]. Por otro lado, el microcontrolador a implementar, requiere de un módulo de transmisión y recepción de datos serial UART para la comunicación con el módulo XBee además de un módulo CCP que sea capaz de generar señales PWM para la conversión digital-análoga de las señales que serán enviadas a los motores.

Por este motivo, se decide utilizar el microcontrolador PIC16F876A, que cumple con éstas características, puesto que posee una memoria de 8192 bytes y 22 terminales de entrada/salida, suficientes para leer las dos señales en cuadratura de cada uno de los encoders y las salidas que estarán conectadas al módulo XBee. Por otro parte tiene un módulo de comunicación serial UART y salidas PWM con una resolución máxima de 10 bits, suficientes para llevar la información correspondiente de las señales de salida del lazo de control:

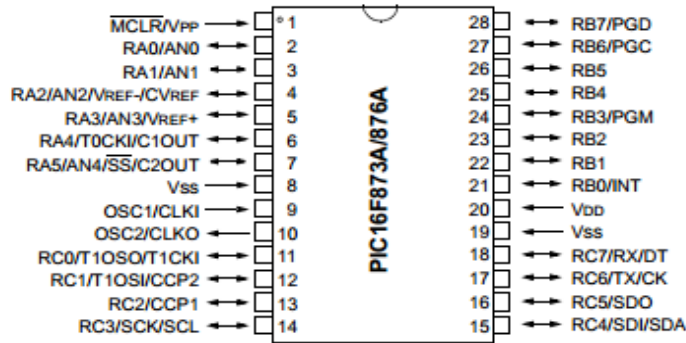


Figura 14. Microcontrolador PIC16F876A

Los pines RA0, RB0, RC0,....., RA5, RB7, RC7 corresponden a las entradas/salidas digitales donde se conectan las señales de los encoders. Específicamente, el pin RC6 y RC7 corresponden al puerto de transmisión y recepción UART respectivamente, que se conectan al puerto de entrada y salida del módulo XBee para la comunicación constante entre estos dos. No obstante, los pines RC0 y RC1 son las salidas que poseen los módulos PWM, de tal modo que son usadas para la conversión de las señales provenientes del control. Puesto que como sólo algunas de ellas se van a conectar, las otras son conectadas a la fuente de alimentación a través de una resistencia, de tal manera que la entrada de ruido al microcontrolador sea mínima.

El pin VDD corresponde a la alimentación del microcontrolador que puede ser entre 2 y 6 voltios. En el caso de la tarjeta de adquisición, el voltaje de alimentación para este dispositivo es de 3.3 V, debido a la alimentación que necesita el módulo XBee y el microcontrolador para su correcto funcionamiento. Sin embargo los pines Vss corresponden a la conexión a tierra y el pin Vpp corresponde a la señal de reset del microcontrolador, que en éste caso es conectado a fuente. Por último los pines OSC1 y OSC2 corresponden a la conexión del oscilador externo o reloj del microcontrolador, cuyo funcionamiento se detallará más adelante [9].

### 6.2.1.1 DESARROLLO EN EL MICROCONTROLADOR

El desarrollo de Software sobre el microcontrolador se realizó por medio del lenguaje de programación CCS, que está orientado a código de C++ por medio de la herramienta de software libre PIC C Compiler. Además, se usó el programa Pickit 3.0 de microchip para la programación de hardware del microcontrolador.

La programación del microcontrolador en la tarjeta de adquisición tiene cuatro fases principales:

- Definición de comunicación serial vía UART.
- Algoritmo de captura, multiplexación y envío de datos provenientes de los encoders del helicóptero.
- Algoritmo de recepción y demultiplexación de datos provenientes de QuaRC®.
- Salida de los datos recibidos por medio de señales PWM.

A continuación, se hará una descripción detallada de cada uno de ellos.

### 6.2.1.1.1 COMUNICACIÓN SERIAL VIA UART

El módulo UART (Universal Asynchronous Receiver and Transmitter) es un dispositivo que incorporan la mayoría de microcontroladores y permite establecer una comunicación de tipo serial entre el microcontrolador y otro dispositivo. Dado que es serial, éste dispositivo se rige por los estándares RS-232, que establece los niveles de voltaje, velocidad de transmisión, bits enviados, etc. Algunas de sus características principales son:

- Transmisión Half-Duplex.
- Modos de operación asíncrona o síncrono.
- Generación de velocidades de transmisión altas con alta resolución.
- Detección de errores por paridad.
- Registros de transmisión y recepción independientes.

En este caso, la comunicación que se realiza entre el módulo XBee USB y el microcontrolador es asíncrona, puesto que el envío de los datos no se sincroniza por una señal de reloj común entre los dos, sino que la sincronización se realiza por medio de un bit de parada a cada byte de datos que se envía y recibe en la comunicación, además la velocidad de transmisión de datos se establece en un principio y es igual entre los dos, que es 38400 baudios (38400 bps) que se escogió debido al ancho de banda que posee el helicóptero, que se puede observar en los diagramas de Bode obtenidos de la respuesta en lazo abierto del anexo 2.

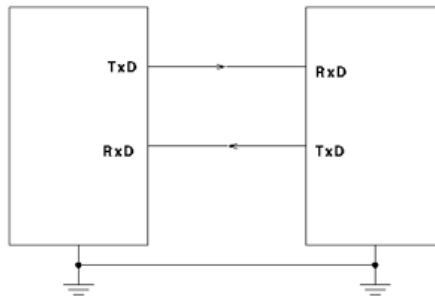


Figura 15. Comunicación serial UART asíncrona.

Mediante esta conexión, fue posible asegurar una alta fiabilidad en la llegada de los datos provenientes de los encoders y del lazo del control, además de una implementación sencilla en la tarjeta, como no ocurre por ejemplo con una comunicación por puerto paralelo, que aumenta el número de conexiones en la tarjeta y el tratamiento de la información en MATLAB® se puede dificultar.

Básicamente el modo de establecer la conexión y comunicación del módulo USART se realiza en CCS por medio de la sentencia `#use rs232()`, la cual activa los pines RC6/TX y RC7/RX para cumplir con las funciones de enviar y recibir datos. Además, con ésta sentencia se configura la velocidad de transmisión de los datos (38400 baudios), el bit de paridad que no es necesario en este caso y el tamaño del paquete de datos que va a enviar y recibir el microcontrolador (8 bits).

Algunos de los registros que utiliza UART en el microcontrolador son: TXREG (Registro para datos de transmisión), RCREG ( Registro de datos recibidos), RCSTA (Registro de control de recepción de datos) y TXSTA(Registro de control de transmisión de datos). [27]

### 6.2.1.1.2 ALGORITMO DE CAPTURA, MULTIPLEXACION Y ENVIO DE DATOS DEL HELICOPTERO

La programación del microcontrolador para la lectura de los encoders se basa en la verificación en todo momento de las señales que entran al microcontrolador y las que se envían a las salidas mediante un bucle o ciclo infinito, que permite de forma sencilla controlar más de un encoder a la vez, al contrario de algún método por interrupciones que no permite de controlar por hardware fácilmente dos o más encoders en las entradas. Por otro lado, la detección de cada giro que genera el motor se asocia a un contador por cada encoder, que se incrementa o decrementa según la dirección de giro del motor y por tanto, definir cuantos grados están relacionados con los ángulos de pitch y yaw. Si el valor de conteo es el máximo de la resolución que posee cada encoder, el contador se coloca en cero y vuelve a empezar.

Para obtener la máxima resolución de los encoders, se utiliza el modo 4X, que multiplica por 4 la resolución original de cada encoder. Mediante este método, se genera un pulso de conteo en cada flanco de subida y de bajada de las señales en cuadratura del encoder A y B, además de determinar la dirección de giro de los motores dependiendo cuál de estas señales está delante de la otra. [28]

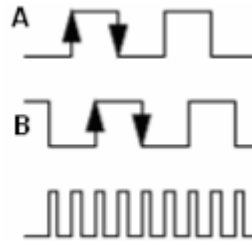


Figura 16. Detección de pulsos por modo 4X

En este caso, se utilizó como referencia para determinar la dirección de giro la secuencia B-A y se parte como base para la programación la siguiente máquina de estados, que indica que cuando la señal A adelanta a B, el motor gira en el sentido contrario a las manecillas del reloj y por tanto el contador se decrementa y en el caso en el que A no adelanta a B, el motor gira en el sentido de las manecillas del reloj y el valor del contador aumenta:

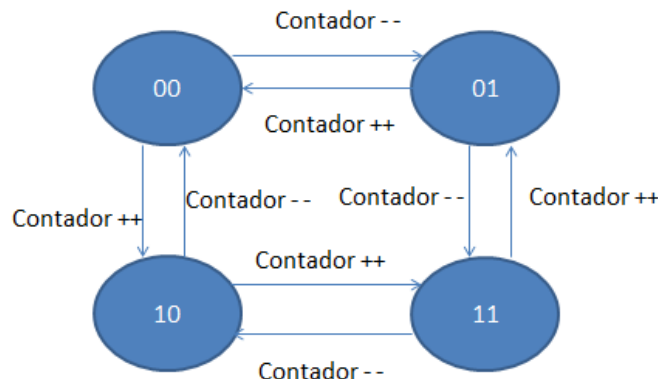


Figura 17. Máquina de estados, conteo de pulsos de encoders

Luego de realizar el conteo de los pulsos provenientes de los encoders, estos datos deben ser enviados al módulo XBee. Dado que se realiza la toma de datos tanto para el encoder de pitch como el de yaw y sólo se tiene un puerto transmisor UART, estos dos datos se multiplexan en tiempo para que no exista algún tipo de colisión o pérdida de información.

Por otro parte, dado que la resolución máxima del ADC que posee internamente el módulo XBee es de 10 bits, no se puede enviar todo el valor de los pulsos de los encoders en un solo paquete, puesto que la máxima resolución que maneja pitch es de 12 bits y la de yaw es 13 bits, por lo que es necesario enviar esta información en dos paquetes de 8 bits. Lo anterior implica que ya no sólo se deben multiplexar dos señales sino cuatro y de forma organizada, puesto que MATLAB® debe capturar estos datos y unirlos de nuevo para generar el valor real de los conteos por lo que se hace estrictamente necesario multiplexar de forma ordenada estos cuatro paquetes.

Para realizar esto, se sigue utilizando el bucle while infinito para enviar los datos de forma constante y se utilizan las funciones make(), que divide en paquetes de ocho bits el valor del contador de pitch y de yaw, y la función putchar(), que envía cada uno de estos por el puerto serial. Estos cuatro paquetes se envían de forma secuencial y tanto el ciclo infinito como la transmisión serial aseguran que los datos llegan al módulo XBee de la misma forma como fueron organizados en CCS. Por último, para que MATLAB® pueda identificar cuáles son los dos bytes de pitch y los dos bits de yaw, se envía primero una bandera con un valor conocido por MATLAB®, y así se garantiza cual contador es pitch y cual es yaw. En el anexo 3 se puede apreciar el código fuente de este algoritmo.

#### **6.2.1.1.3 ALGORITMO DE RECEPCION Y DEMULTIPLEXACION DE LOS DATOS PROVENIENTES DE QUARC®**

Otra de las funciones que cumple el microcontrolador es recibir los datos de salida del lazo de control, que corresponden a niveles de voltaje que se envían a los motores por medio de un circuito de acondicionamiento y los módulos de potencia. Para lograr esto, nuevamente se utiliza la comunicación serial vía UART, en éste caso, con el puerto de recepción de datos.

Con el fin de asegurar que el envío y recepción de los datos sea constante y que uno no afecte el otro, se utiliza para la recepción de los datos una interrupción, en este caso la interrupción de recepción por puerto serial RDA\_isr(), que detiene la ejecución del programa principal y lo reactiva cuando su rutina finaliza, de tal modo que no se pierde información en la transmisión o en la recepción. Dentro de esta interrupción se utiliza la sentencia getch(), que obtiene los datos provenientes del módulo XBee.

Es importante tener en cuenta que Simulink-QuaRC® multiplexa las señales de salida del control por un tiempo igual que el microcontrolador en el envío, por lo que se hace necesario demultiplexar los datos en ésta interrupción cada vez que los capture y enviar pitch y yaw de forma independiente hacia las dos salidas de PWM que posee el microcontrolador. Para realizar esto, se multiplexa una bandera constante de valor cero en Simulink a las otras dos señales, de tal forma que cuando el microcontrolador reciba esta señal, se vuelva a recibir datos, etiquetándolos como pitch y yaw y mandarlos a las salidas PWM. Esto se logra gracias a que Simulink asegura que el orden en el que se multiplexan las señales es el mismo al que se envía.

A continuación se presenta un diagrama de flujo con el algoritmo de recepción de datos:



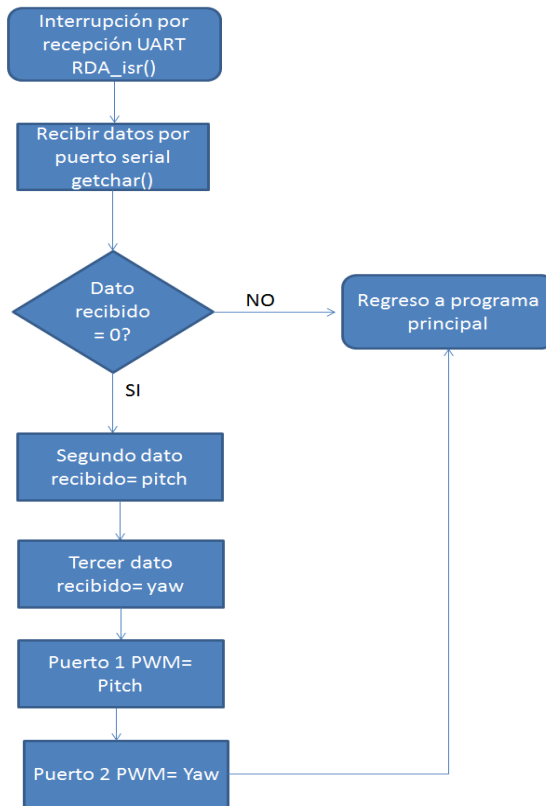


Figura 18. Diagrama de flujo de algoritmo de recepción de datos

#### 6.2.1.1.4 SALIDA DE DATOS POR MEDIO DE SEÑALES PWM

Luego de realizar la demultiplexación de las señales de salida de pitch y yaw, es necesario enviar estos valores por medio de señales de tipo PWM para luego convertirlas en señales análogas DC y ser enviadas a los motores del helicóptero. La configuración de las señales PWM se realiza mediante los módulos CCP que posee el microcontrolador utilizado y debe tener las siguientes consideraciones:

- Configuración del temporizador con el que opera el módulo CCP
- Resolución de la señal de PWM
- Frecuencia de la señal.
- Configuración de ancho de pulso.

En el caso del microcontrolador PIC16F876A, el temporizador que debe ser configurado para el correcto funcionamiento del módulo CCP es el timer2, que en el programa principal debe ser configurado como:

```
setup_timer2(T2_DIV_BY_Prescaler,PR2, Postcaler)
```

Siendo T2\_DIV\_BY\_Prescaler un parámetro constante que se halla a partir de la frecuencia de la señal de PWM, PR2 el valor de desborde del temporizador que se calcula con el ancho de pulso de la señal y Postcaler una constante de valor 1 que no interviene en el funcionamiento del módulo.

Teniendo en cuenta que la resolución es de 10 bits para la señal de PWM, los cálculos a partir de los cuales se obtienen estas constantes son los siguientes:

$$Resolución\ PWM = \frac{\log\left(\frac{F_{osc}}{F_{pwm}}\right)}{\log(2)}$$

$$F_{PWM} \frac{F_{osc}}{10^{Resolución\ PWM * \log(2)}} = \frac{20\ MHz}{1024} = 19.53\ kHz$$

$$T_{PWM} = 51.67\ \mu s$$

$$T_{PWM} = [PR2 + 1] * 4 * T_{osc} * Prescaler$$

$$Ancho\ de\ pulso = \frac{x}{PR2 + 1}$$

Como se requiere que para un 100% del ancho de pulso de la señal de PWM el valor de x que corresponde a la señal de pitch o yaw sea de 256 (resolución de 8 bits), entonces:

$$PR2 = 255$$

Por tanto, de la ecuación de periodo de PWM se obtiene el valor de T2\_DIV\_BY\_Prescaler, que corresponderá a una constante que divide la frecuencia del microcontrolador entre su valor:

$$Prescaler = 1$$

Luego de realizar la configuración del timer2, en la interrupción donde se realiza la recepción y demultiplexación de las señales de pitch y yaw, cada una de estas se asigna a los dos módulos CCP del microcontrolador por medio de las sentencias set\_pwm1\_duty() y set\_pwm2\_duty() respectivamente, las cuales se definen con el valor que se recibe de QuaRC® y que determina el ancho de pulso de estas señales de PWM.

La señal de PWM está en el rango de 0 V a 3.3 V, debido a la polarización del microcontrolador, lo que significa que para un ciclo útil del 0%, la señal de salida será de 0 V y para un ancho de pulso del 100% será de 3.3 V. Es importante aclarar que el microcontrolador entrega un ciclo útil máximo del 99.6%.

## 6.2.2 OSCILADOR

El microcontrolador PIC16F876A requiere de un circuito externo que le indique a qué velocidad debe trabajar, como lo hace un oscilador, que genera pulsos de reloj a una frecuencia determinada, que depende de la precisión y velocidad que se requiere. En éste caso se utilizará un oscilador de 20 MHz, que es suficiente para la velocidad de transmisión de los datos de los encoders y garantiza un buen arranque del microcontrolador, además genera la máxima resolución de las señales de PWM. Es claro tener en cuenta que internamente, la frecuencia del oscilador se divide en cuatro, por lo que la frecuencia de trabajo real del microprocesador es de 5 MHz [9].

Este microcontrolador trabaja con cuatro modos distintos de oscilador externo que se debe especificar en el algoritmo de programación, los cuales son:

- **LP:** Cristal para baja frecuencia y bajo consumo de potencia
- **XT:** Cristal
- **HS:** Cristal de alta velocidad
- **RC:** Oscilador con resistencia y condensador

En la tarjeta de adquisición de datos se usará el modo HS, puesto que las frecuencias de oscilación posibles se encuentran entre 4 MHz y 20 MHz, suficientes para la implementación de un cristal de 20MHz y por ende, una buena velocidad de trabajo del microcontrolador para la lectura de los datos de los encoders. Además, con un cristal se asegura una mayor precisión en el valor de frecuencia que arroja como no ocurre con osciladores RC. La conexión típica del oscilador en el microcontrolador es la siguiente:

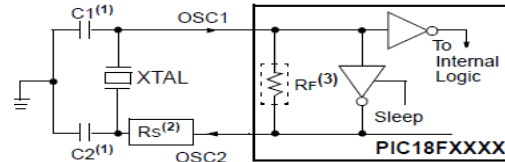


Figura 19. Oscilador de cristal conectado a microcontrolador

A una frecuencia de 20 MHz en el cristal, los valores de los dos condensadores que se conectan en las terminales de éste pueden tomar valores entre 15 y 68 pF. En casos especiales o para tener una mayor precisión se utiliza una resistencia serie  $R_s$  [9]. En la tarjeta de adquisición se utilizaron dos condensadores de 22 pF.

### 6.2.3 MÓDULOS XBEE/XBEE-PRO OEM RF

El módulo XBee fue diseñado bajo el estándar IEEE 802.15.4. Trabaja en la banda de frecuencia de los 2.4GHz con una velocidad de transmisión de datos de 250 kbps y una resolución de 8 bits que permite realizar la transmisión y recepción de datos de manera confiable. En la tabla 3 se describe brevemente la función de cada pin:



Figura 20. Módulo XBEE

Tabla 3. Pines correspondientes al módulo XBee [14]

PIN	Nombre	Descripción
1	VCC	Voltaje de alimentación del circuito entre 2.8 y 3.4 VDC
2	DOUT	Salida de Datos UART para conexión con XBee Explorer USB
3	DIN/CONFIG	Entrada de Datos UART para conexión con XBee Explorer USB
4	DO8	Salida Digital 8
5	~RESET	Reinicio del Módulo (Normalmente Bajo)
6	PWM0/RSSI	Salida del PWM0 (Configurable) ó Indicador de intensidad de la señal
7	PWM1	Salida del PWM1 (Configurable)
8	Reservado	Sin conexión
9	~DTR/SLEEP_RQ/DI8	Función Sleep ó entrada digital 8
10	GND	Tierra
11	AD4/DIO4	Entrada Análoga 4 ó I/O Digital 4
12	CTS/DIO7	Control de flujo de la señal Clear-to-Send ó I/O Digital 7
13	ON/SLEEP	Indicador de estado del Módulo (encendido/reposo)
14	VREF	Voltaje de Referencia para entradas A/D
15	Associate/AD5/DIO5	LED Indicador asociado al módulo, Entrada Análoga 5 ó I/O Digital 5

<b>16</b>	~RTS/AD6/DIO6	Control de Flujo de la señal Ready-to-Send, Entrada Análoga 6 ó I/O Digital 6
<b>17</b>	AD3/DIO3	Entrada Análoga 3 ó I/O Digital 3
<b>18</b>	AD2/DIO2	Entrada Análoga 2 ó I/O Digital 2
<b>19</b>	AD1/DIO1	Entrada Análoga 1 ó I/O Digital 1
<b>20</b>	AD0/DIO0	Entrada Análoga 0 ó I/O Digital 0

Todas la entradas y salidas pueden ser configuradas con los comandos AT y/o la aplicación API como se explica en la hoja de especificaciones del Módulo XBee [14].

### 6.2.3.1 PINES QUE SOPORTAN MÚLTIPLES FUNCIONES

Algunos pines del módulo XBee soportan más de una función. Para acceder a estas funciones se debe configurar el modulo usando los comandos AT y las funciones de soporte API (Application Programming Interface) en el momento de hacer la configuración.

Los pines 20, 19, 18, 17, 15 y 11 tienen como función adicional un conversor Análogo/Digital cada uno, los cuales pueden habilitarse fácilmente utilizando solo un comando AT. Los pines 9, 12, 15, 16 y 17 tienen otras funciones de soporte que se habilitan con API y se muestran a continuación.

**~DTR:** Data Terminal Ready (Normalmente Bajo), es una señal que puede activan el apagado del módulo este pin está relacionado con la función de Sleep (reposo) ya que la única línea que puede sacar al módulo del estado de Sleep.

**SLEEP\_RQ:** Esta función permite “despertar” de su modo de reposo, para hacerlo debe negarse la señal SLEEP\_RQ.

**CTS:** Cuando el buffer de entrada de datos está cerca del límite de memoria (solo quedan 17 bytes libres) el módulo niega Clear-to-Send (~CTS) enviando así una señal al host para que deje de enviar datos. Cuando el buffer tiene 34 bytes de memoria disponible el módulo pone en alto CTS y los datos pueden seguir entrando.

**Associate:** Es una LED que indica si hay algún modulo asociado transmitiendo, esta función solo puede ser activada cuando se conectan los módulos en modo de operación NonBeacon.

**~RTS:** Ready-to Send (Normalmente Bajo). Si el control de flujo de la señal ~RTS se habilita los datos no serán enviados al buffer de salida de datos DO hasta que ~RTS se ponga en alto.

Existen dos formas de comunicación de los módulos XBee:

Una es el **modo transparente** en la cual el módulo se comporta como un "cable inalámbrico" de forma que el puerto serie puede ser transmitido como si fuese un cable. Esta es la forma más sencilla donde se puede conectar dos dispositivos sincronizando su velocidad mediante una aplicación que permita configurar los XBee. Para ello MaxStream pone a disposición una herramienta software gratuita (X-CTU).

La segunda forma es más compleja, ya que utiliza la **API** que incorpora el microcontrolador de cada módulo XBee. En esta modalidad se puede generar una red de módulos XBee y programar un XBee para que trabaje como repetidor de señal (hub), además de otras funciones que permiten tener ventajas en los protocolos de comunicación. Una de las funciones más importantes es que el módulo que se emplea como emisor/receptor "dice" si el módulo al que se quiere comunicar está disponible, es decir, devuelve una respuesta cada vez que se intenta comunicar con otro módulo. Toda la comunicación se hace por medio de comandos AT.

Para la realización de este proyecto se emplea una conexión tipo NonBeacon entre los módulos XBee y el modo de comunicación transparente. Los sistemas NonBeacon operan dentro de una topología de red Peer-to-Peer (Punto a Punto), por lo que no dependen de las relaciones maestro/esclavo. Esto significa que los módulos permanecen sincronizadas sin el uso de configuraciones maestro/servidor y cada módulo de la red comparte tanto el rol de maestro como el de esclavo. La arquitectura Peer-to-peer cuenta con rápidos tiempos de sincronización y rápidos tiempos de arranque en frío.

### 6.2.3.2 TARJETA XBEE EXPLORER USB

Para poder enviar los datos que el módulo XBee recibe y también proporcionarle datos para que transmita es necesario conectar el modulo al computador. Esto se hace mediante la tarjeta XBee Explorer USB, la cual convierte las señales digitales provenientes del XBee a datos en forma serial que se envía a través de USB al computador para poder ser procesados en MATLAB® e implementar el controlador.



Figura 21. Tarjeta XBee Explorer USB

La tarjeta XBee Explorer está compuesta de un puerto USB, dos conectores PPPN101BF para insertar el módulo XBee y el circuito integrado FT232RL. Este es una interfaz UART USB a serial que permite convertir los datos provenientes del XBee a Serial. En la Figura 22 se muestra la configuración de para el circuito integrado.

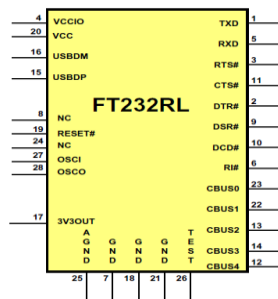


Figura 22. Circuito integrado FT232RL

En la tabla 4 se explica la conexión de los pines con relación al módulo XBee:

Tabla 4. Conexión de los pines para el Circuito Integrado FT232RL [15]

Pin FT232RL	Pin Módulo XBee	Descripción
1-TXD	3-DIN	Pin de entrada de datos del módulo Zigbee al pin de transmisión del FT232RL
5-RXD	2-DOUT	Pin de salida de datos del módulo Zigbee al pin de recepción del FT232RL
3--RTS	16--RTS	Conexión directa entre pines Ready to Send
11--CTS	12--CTS	Conexión directa entre pines Clear to Send
2--DTR	9-DTR	Conexión directa entre pines Data Terminal Ready
7,18,21,25,26	10	GND

Los pines 15 y 16 van conectados directamente al puerto USB. Estos pines envían y reciben los datos de manera serial al computador y desde el mismo. Es importante tener en cuenta que la tarjeta XBee Explorer se alimenta con un voltaje de polarización de +5 VDC, pero el módulo Zigbee tiene un voltaje de polarización máximo de 3.4 VDC, es por eso que se utiliza un regulador de voltaje a +3.3 VDC para cumplir con las especificaciones del módulo.

### 6.2.3.3 CONFIGURACION DEL MODULO XBEE EN LA TARJETA DE ADQUISICION

Los módulos XBee que se adquirieron pertenecen a la serie 1 de XBee Digi International, ésta serie en particular está basada en el estándar 802.15.4 pero tiene algunas modificaciones que mejoran su rendimiento, a continuación se muestra una tabla que permite ver las diferencias entre XBee serie 1 y la serie 2.

Tabla 5. Comparación de características entre XBee serie 1 y serie 2 [21]

	<b>XBee Series 1</b>	<b>XBee Series 2</b>
<b>Indoor/Urban range</b>	Up to 100 ft. (30m)	Up to 133 ft. (40m)
<b>Outdoor RF line-of-sight range</b>	Up to 300 ft. (100m)	Up to 400 ft. (120m)
<b>Transmit Power Output</b>	1 mW (0dbm)	2 mW (+3dbm)
<b>RF Data Rate</b>	250 Kbps	250 Kbps
<b>Receiver Sensitivity</b>	-92dbm (1% PER)	-98dbm (1% PER)
<b>Supply Voltage</b>	2.8 - 3.4 V	2.8 - 3.6 V
<b>Transmit Current (typical)</b>	45 mA (@ 3.3 V)	40 mA (@ 3.3 V)
<b>Idle/Receive Current (typical)</b>	50 mA (@ 3.3 V)	40 mA (@ 3.3 V)
<b>Power-down Current</b>	10 Ua	1 uA
<b>Frequency</b>	ISM 2.4 GHz	ISM 2.4 GHz
<b>Dimensions</b>	0.0960" x 1.087"	0.0960" x 1.087"
<b>Operating Temperature</b>	-40 to 85 C	-40 to 85 C
<b>Antenna Options</b>	PCB, Integrated Whip, U.FL, RPSMA	PCB, Integrated Whip, U.FL, RPSMA
<b>Network Topologies</b>	Point to point, Star, Mesh (with DigiMesh firmware)	Point to point, Star, Mesh
<b>Number of Channels</b>	16 Direct Sequence Channels	16 Direct Sequence Channels
<b>Filtration Options</b>	PAN ID, Channel & Source/Destination	PAN ID, Channel & Source/Destination

XBee serie 1 viene con el firmware 802.15.4 estándar para topologías punto-punto o de estrella. Este firmware ofrece entradas ADC (análogo-digital) y I/O digital y analoga. El 802.15.4 XBee es significativamente más rápido que ZigBee; la latencia de RF generalmente se puede calcular en 802.15.4. El rendimiento es también mucho mayor, en la práctica un rendimiento máximo es de alrededor de 80kbps [21].

XBee serie 2 no ofrece ningún firmware 802.15.4, sino que siempre se está ejecutando el firmware malla ZigBee. Los ZigBee XBee sobresalen en escenarios de muy baja potencia, cuando se configuran como

dispositivo final (end Device), este módulo tiene el menor consumo de corriente de cualquier producto de Digi RF. Sin embargo, la infraestructura de una red ZigBee es más compleja y requiere más configuración para implementarla completamente [21].

En el caso del desarrollo de este trabajo de grado solo es necesaria la implementación de una red punto a punto non beacon, es decir que ninguno de los dispositivos esté ligado a una configuración maestro-esclavo o router-coordinador, es por esta razón que se utiliza XBee serie 1.

A continuación se mostrara una tabla con los valores configurados para cada uno de los módulos, para realizar esta operación se utilizó el software X-CTU, software propietario de MaxStream.

Tabla 6. Configuración de los módulos XBee.

	<b>Parámetros</b>	<b>XBee Base</b>	<b>XBee Remoto</b>
<b>Networking &amp; Security</b>	CH-Channel	B	B
	ID- PAN ID	3456	3456
	DH-Destination Address High	0	0
	DL- Destination Address Low	2	1
	MY- 16-bit Source Address	1	2
	SH - Serial Number High	N/A	N/A
	SL - Serial Number Low	N/A	N/A
	RN - Random Delay Slots	N/A	N/A
	MM - MAC Mode	N/A	N/A
	CE - Coordinator Enable	N/A	N/A
	SC - Scan Channels	N/A	N/A
	SD - Scan Duration	N/A	N/A
	A1 - End Device Association	N/A	N/A
	A2 – Coordinator Association	N/A	N/A
	AI - Association Status	N/A	N/A
<b>RF Interfacing</b>	PL - Power Level	4-Highest	4-Highest
	CA - CCA Threshold	N/A	N/A
<b>Serial Interfacing</b>	BD - Interface Data Rate	5-38400	5-38400
	NB-Parity	0 - None	0 – None
	RO – Packetization Timeout	1	1
	AP-API Enable	0 - Disabled	0 - Disabled

Los parámetros que no se muestran en la tabla 6 se dejaron por defecto. Para conocer el proceso de configuración del módulo XBee con la herramienta X-CTU referirse al anexo 13.

#### 6.2.4 FILTRO PASA BAJOS ANALOGO

Las salidas del microcontrolador por las cuales se envían las señales de control hacia los módulos de potencia UPM son de tipo PWM (Pulse Width Modulation), que generan básicamente formas de onda digitales, pero que pueden ser usados como sencillos conversores análogos digitales (D/A) usando filtros pasabajos análogos. Lo anterior se basa en lo siguiente:

Una señal típica de PWM tiene una frecuencia fija y un ancho de pulso variable [16]. Este ancho de pulso es directamente proporcional a la amplitud de la señal original:

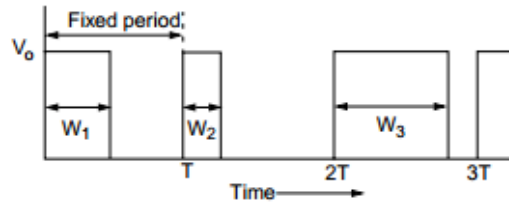


Figura 23. Señal típica de PWM

El análisis de Fourier de la señal de PWM muestra que la componente fundamental de frecuencia es  $F_n = 1/T$  y los armónicos presentes tienen frecuencias  $F = K/T$ , siendo K un entero. Estos armónicos requieren ser eliminados, por lo que se usa un filtro pasabajos análogo.

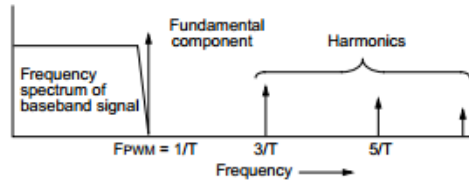


Figura 24. Espectro de la señal de PWM

Para el diseño del filtro en la tarjeta de adquisición se tiene en cuenta que la señal que se debe enviar a los módulos de potencia debe ser análoga DC, puesto que los motores requieren este tipo de señal para ser manejados. Dado que la frecuencia de PWM de los módulos CCP del microcontrolador calculada anteriormente es de 19.531 kHz, el filtro debe eliminar ésta componente de frecuencia y además tener una frecuencia de corte muy baja para entregar a los módulos una señal que sea prácticamente un DC, es decir con un rizado de voltaje muy pequeño y que no modifique las amplitudes correspondientes a las señales de control originales.

Usualmente se utilizan filtros de primer orden pasivos RC para generar una señal análoga a partir de una señal de PWM, pero que presenta problemas como lo son un ripple alto en la banda de paso y un efecto de Roll-off en la banda de rechazo, que en el caso de las señales que se quieren manejar para el control de los motores, pueden ser susceptibles ante éstos efectos y el rendimiento del sistema no será el esperado.

Por tanto, se decidió usar un filtro activo de orden superior que genera mayor rendimiento frente a éstos problemas, teniendo en cuenta que un orden muy alto genera mayor complejidad en el cálculo y en la implementación del circuito. Es por esto que se optó por usar un filtro pasabajos activo Butterworth de tercer orden, que genera una mínima atenuación y rizado en la banda de paso y que la caída en magnitud de la banda de rechazo lo hace más rápido que otro tipo de filtro de éste tipo.

La función de transferencia del filtro a utilizar es la siguiente:

$$\frac{V_{out}}{V_{in}} = \frac{1}{[s^2 + s\left(\frac{1}{C_1R_2} + \frac{1}{C_1R_3}\right) + \frac{1}{R_2R_3C_1C_2}] + [s + \frac{1}{R_1C_3}]}$$

La frecuencia de corte del filtro se define por las siguientes ecuaciones:

$$f_c = \frac{1}{2\pi\sqrt{R_2R_3C_1C_2}}, \quad f_c = \frac{1}{2\pi R_1C_3}$$

Dado que los componentes del filtro serán implementados en montaje superficial y serán de precisión, se eligen los siguientes valores a una frecuencia de corte de 5 Hz:

$$R_1 = 31.83 \text{ k}\Omega, R_2 = 22 \text{ k}\Omega, R_3 = 9.09 \text{ k}\Omega, C_1 = 5.6 \text{ }\mu\text{F}, C_2 = 1.2 \text{ }\mu\text{F}, C_3 = 1\text{ }\mu\text{F}$$



El circuito obtenido es el siguiente, teniendo en cuenta que se escogió un amplificador operacional MC33204 que tiene ancho de banda de 2.2 MHz, es rail to rail, con un slew rate de 1 V/ $\mu$ s y puede ser polarizado entre  $\pm 0.9$  V y  $\pm 6.0$  V [17]:

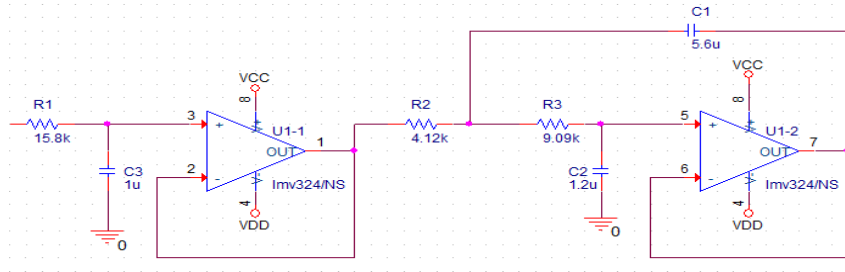


Figura 25. Filtro pasabajos de tercer orden

En el anexo 4 del documento se muestra la respuesta del filtro en magnitud y en fase.

### 6.2.5 ACONDICIONAMIENTO DE NIVEL

Al realizar pruebas con el helicóptero se observó que las señales de control enviadas a las UPM, tomaban tanto valores negativos como positivos, por lo tanto es necesario implementar un restador a cada una de las señales de PWM que se tienen a las salidas del microcontrolador con el fin de bajarlas un nivel DC de 1.65V y así obtener unas señales centradas en 0 y con valores entre  $\pm 1.65$ V.

A continuación en la figura 26 se muestra un esquemático del restador implementado con un amplificador operacional de ganancia 1. [24]

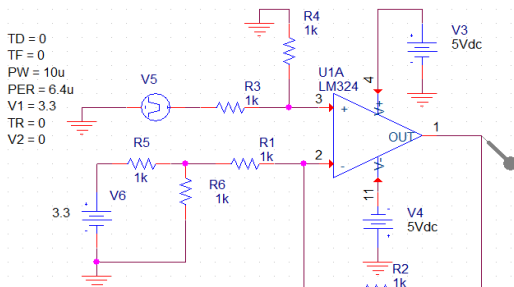


Figura 26. Esquemático del restador para bajar el nivel de voltaje DC de las señales PWM.

### 6.2.6 ACONDICIONAMIENTO DE SEÑAL

Las señales que deben recibir las UPM deben estar entre  $\pm 5$  VDC, para esto es necesario amplificar la señal recibida de los filtros antes de enviarlas a las UPM, para esto se utiliza la siguiente configuración de un amplificador no inversor con ganancia 3.0.

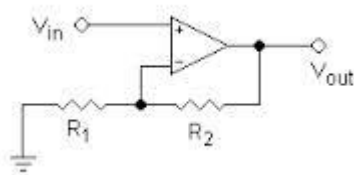


Figura 27. Configuración para amplificador no inversor

Para calcular la ganancia:

$$A = 1 + \left(\frac{R2}{R1}\right) \quad [25]$$

Donde,

$$R1 = 15k\Omega \text{ y } R2 = 30k\Omega$$

### 6.3 MODULOS DE POTENCIA (UPM)

Los módulos de potencia UPM son amplificadores de potencia que se requieren para el control de los motores del helicóptero 2DOF y sus características principales son: alimentación de  $\pm 12\text{ V}$ , entradas análogas y salida análoga amplificada, donde la ganancia depende del cable que se conecte a las terminales de la UPM. Las conexiones típicas de una UPM son las siguientes:

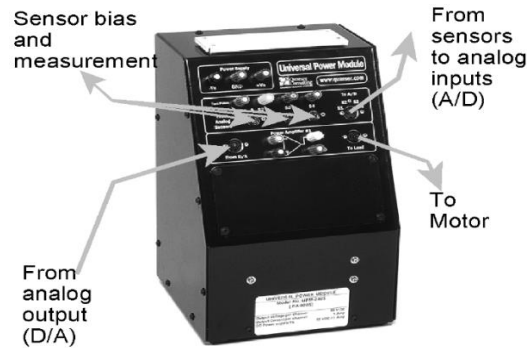


Figura 28. Conexiones típicas en UPM

La señal que se conecta a la entrada de las UPM debe ser análoga y corresponde a la señal de control que proviene de las salidas PWM del microcontrolador de la tarjeta de adquisición de datos y que ha sido convertida de señal digital a señal análoga. Esta señal se conecta a la UPM por medio de un cable RCA a 5-pin-DIN.

Por otro lado, la señal de salida amplificada que se conecta a los motores también es análoga, en el caso de éste diseño, se requieren dos UPM que conecten las dos señales de control de los ángulos pitch y yaw a cada uno de sus motores respectivamente. Para el motor de pitch se requiere la conexión de una UPM2405, que significa que posee un máximo voltaje de salida de  $24\text{ V}$  y una corriente máxima de  $5\text{ A}$  y para el motor de yaw una UPM1503 que tiene la misma convención que la otra.

Las señales de salida de la UPM que controlan los motores de pitch y yaw, son conectadas mediante cables 4 pin-stereo-DIN a 6 pin-stereo-DIN que determinan la ganancia del amplificador. Como se dijo anteriormente, para el motor de pitch se conectará un cable de ganancia 5 y para el motor de yaw un cable de ganancia 3 [19]. La forma como a través del cable se puede determinar la ganancia del amplificador es la siguiente:

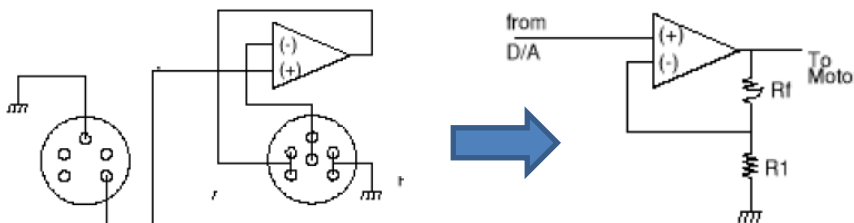


Figura 29. Conexión de cables para amplificación en UPM

Como se puede observar en la figura anterior, los terminales del cable de salida al conectarse con el amplificador operacional forman 2 resistencias que generan un amplificador no inversor con ganancia

$1 + \frac{R_f}{R_1}$ , que dependiendo de la resistencia que se forman en los terminales del cable, generan el valor de ganancia del amplificador.

## 6.4 ADQUISICION Y PROCESAMIENTO DE DATOS EN MATLAB®

El procesamiento y ejecución del controlador en tiempo real y la adquisición de datos de este, se efectúa por medio del módulo QuaRC® que posee Simulink y en el cual se puede adquirir datos ya sea de las tarjetas que posee o de algún puerto en específico del computador.

Como es el caso, los datos provenientes de la tarjeta de adquisición son tomados por medio del módulo XBee USB, los datos deben ser enviados en QuaRC® a través de un bloque de comunicaciones que pueda establecer una conexión cliente servidor entre los datos que envía QuaRC® al compilar el algoritmo y los datos que recibe de la tarjeta de adquisición. A continuación se hará una descripción detallada de la comunicación empleada en QuaRC®.

### 6.4.1 BLOQUE STREAM SERVER DE QUARC®

Para establecer la comunicación del puerto serial del computador con la tarjeta de QuaRC® se utilizó el bloque Stream server de la librería de QuaRC® en Simulink, de esta forma se pueden tomar los datos directamente del XBee y descargarlos en la tarjeta de QuaRC® sin necesidad de ningún otro tipo de software que intervenga en el proceso.

Este bloque se puede encontrar en la librería de Simulink QuaRC® Targets/Communications/Basic. [22]

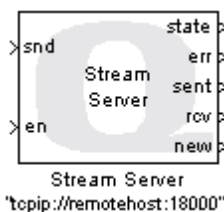


Figura 30. Bloque Stream Server de QuaRC® para Simulink [22]

El bloque Stream Server escucha por conexiones de hosts remotos. Cuando un host remoto intenta conectarse, se acepta la conexión y establece una conexión “persistente” con ese host. Sólo una conexión es aceptada a la vez. El host puede ser local o remoto. Si la conexión con el host se pierde este bloque acepta automáticamente una nueva conexión. El estado actual de la conexión está disponible en la salida **state** del bloque

Cuando el bloque se configura para recibir datos y llegan datos, la salida **new** se pone en verdadero y los datos recibidos se pueden ver en la salida **rcv**. La cantidad de datos que se reciben está determinada por las dimensiones del parámetro *Default output value* y el formato de los datos especificado en el parámetro *Output data type*. Este bloque soporta señales multidimensionales y señales de bus.

Cuando el bloque está configurado para enviar datos, la señal en la entrada **snd** es enviado al host cada instante de muestreo en el que la entrada **en** es distinto de cero. Esta señal puede ser multidimensional o una señal de bus. La salida **sent** se pone en un valor verdadero (distinto de cero) si los datos se han enviado. Si la operación de envío se bloquea la salida **sent** se establece en falso (cero). Si se produce un error entonces un código de error negativo de QuaRC® se emite en la salida **err**. Una condición de error hará que el bloque cierre la conexión e intente aceptar una nueva conexión.

Los hosts remotos se identifican por un identificador de recursos universal (URI), QuaRC® utiliza URI para todas sus comunicaciones, ya que proporciona un medio uniforme, extensible y flexible para identificar el protocolo de comunicación a utilizar y los parámetros de comunicación asociados.

## ➤ CONFIGURACIÓN Y PARÁMETROS DEL BLOQUE

A continuación en las tablas se hará una breve descripción de cada uno de los parámetros configurables del bloque con sus valores respectivos.

Para conocer más a fondo las especificaciones completas de los parámetros de cada una de las pestañas del bloque por favor referirse al anexo 5.

Tabla 7. Pestaña principal de los parámetros configurables para el bloque Stream Server de QuaRC® [22].

PESTAÑA PRINCIPAL		
NOMBRE	DESCRIPCIÓN	VALOR
<b>Source of URI</b>	Indica si el URI debe determinarse a partir de los parámetros del diálogo o un puerto de entrada.	- <i>Specify via dialog</i> - <i>URI upon which to listen</i>
<b>URI upon which to listen</b>	Identifica el URI que se debe escuchar. Este parámetro identifica el protocolo de comunicación y los parámetros asociados.	<i>Serial, TCP/IP, UDP, File, Pipe, Com20020X, Shared Memory, SPI, etc...</i>
<b>Send buffer size in bytes</b>	Indica el tamaño de la memoria intermedia utilizada para el flujo de envío de datos.	- <i>Mayor o igual al vector máximo que será enviado.</i>
<b>Receive buffer size in bytes</b>	Indica el tamaño de la memoria intermedia utilizada para el flujo de recepción de datos.	- <i>Mayor o igual al vector máximo recibido.</i>
<b>Byte Ordering</b>	El orden en que se transmiten los bytes.	- <i>Little Endian, Big Endian</i>
<b>Optimize for</b>	Esta opción controla si los datos a transmitir se vacían en la secuencia subyacente.	- <i>Maximum throughput</i> - <i>Minimum Latency</i>
<b>Implementation</b>	Esta opción controla cómo las comunicaciones subyacentes son implementadas por el bloque.	- <i>Use non-blocking I/O</i> - <i>Use blocking I/O in a separate thread</i>
<b>Send Options</b>	Este parámetro determina si todos los datos que aparecen en la entrada del <b>snd</b> se transmiten, o sólo los datos más recientes o ninguno en absoluto.	-
<b>Recieve Options</b>	Este parámetro determina si todos los datos que se reciben aparecen en la salida <b>rcv</b> , o sólo los datos más recientes, o si los datos se reciben en absoluto.	-
<b>Send FIFO size in samples</b>	El tamaño de la cola FIFO utilizada para transferir los datos a ser enviados desde la entrada <b>snd</b> al hilo enviado cada instante de muestreo.	- <i>Desde 0 hasta 4.294.967.296 muestras, de tal forma que los datos sean recogidos mientras el envío de los datos anteriores está en proceso. (Ver anexo 5)</i>
<b>Receive FIFO size in samples</b>	El tamaño de la cola FIFO utilizada para transferir los datos recibidos por el hilo receptor a la salida <b>rcv</b> del bloque cada instante de muestreo.	- <i>Desde 0 hasta 4.294.967.296 muestras de tal forma que los datos recibidos sean almacenados hasta el siguiente instante de muestreo. (Ver anexo 5)</i>
<b>Send thread priority</b>	La prioridad del hilo de envío cuando la aplicación multiproceso se utiliza.	- <i>0 indica la prioridad más baja, valores mayores a 0 indican mayor prioridad.</i>
<b>Receive thread priority</b>	La prioridad del hilo recetor cuando la aplicación multiproceso se utiliza	- <i>0 indica la prioridad más baja, valores mayores a 0 indican mayor prioridad.</i>
<b>Default output value</b>	El valor de salida desde la salida <b>rcv</b> antes de que cualquier dato se haya recibido. Este parámetro también determina el tamaño de la salida <b>rcv</b> .	-
<b>Sample time</b>	El tiempo de muestreo del bloque	- <i>Tiempo Continuo : 0</i> - <i>Tiempo discreto : &gt;0</i> - <i>Tiempo Heredado: -1</i>

Tabla 8. Pestaña secundaria de los parámetros configurables para el bloque Stream Server de QuaRC® [22].

PESTAÑA SIGNAL DATA TYPES		
NOMBRE	DESCRIPCIÓN	VALOR
<b>Output data type</b>	Determina el tipo de formato de los datos recibidos	<i>-Back propagation, Specify via dialog</i>
<b>Output type name</b>	Este parámetro debe ser el nombre de una variable en el espacio de trabajo de MATLAB® que define el tipo de datos de la salida.	-

### URI para protocolo serial:

El Identificador de Recursos Universal o URI se utiliza para identificar los recursos, QuaRC® hace uso extensivo de los URI. [23]

En el caso de este trabajo de grado se utilizara un protocolo de comunicación serial por lo cual es necesario utilizar un URI que permita este tipo de comunicación.

Los URI en QuaRC® tienen la siguiente forma:

<protocol>:<hier-part>?<option>=<value>,...

Para el protocolo serial específicamente se muestra un ejemplo a continuación:

serial://localhost:2?baud=57600,word=8,parity=none,stop=1

En el ejemplo se observan las siguientes características, localhost:2 se refiere a que va a hacer su conexión con un host local a través del puerto COM 2, después del signo de interrogación (?) se configuran los parámetros del puerto COM 2 para poder realizar la transmisión y/o recepción de datos con el Bloque Stream Server, baud=57600 establece la velocidad de transmisión a 57600 Baudios, word=8 establece los 8 bits de datos, Parity=none establece los datos sin paridad y stop=1 da el bit de parada en 1.

En general los parámetros que se pueden configurar con el URI para protocolo serial se muestran en la siguiente tabla:

Tabla 9. Parámetros configurables para el URI del protocolo serial. [23]

Parámetro	Descripción	Valores
<b>Baud</b>	Velocidad de transmisión deseada.	75,110,134,150,300,600,1200,1800,2400,4800,7200,9600,14400,19200,38400,57600,115200,128000,230400,960800,921600
<b>Word</b>	Tamaño de la palabra deseada en bits	5, 6, 7 u 8
<b>Stop</b>	Bit de Parada	1, 1.5 y 2
<b>Parity</b>	Bit de Paridad	None, even, odd, mark y space
<b>Flow</b>	Control de Flujo	None, Hardware, Software
<b>Memsiz</b>	Tamaño de los búfer serial receptor y transmisor	1200 (valor por defecto)
<b>Sndsiz</b>	Tamaño del búfer serial transmisor	1200 (valor por defecto)
<b>Rcvsiz</b>	Tamaño del búfer serial receptor	1200 (valor por defecto)
<b>Device</b>	Prefijo que se utilizará para el nombre del dispositivo serie	-

## 7. CIRCUITO IMPRESO

A continuación se muestra el diseño del PCB para la tarjeta de adquisición de datos, el cual fue realizado mediante el software Eagle. En la figura 31 se muestra la capa BOTTOM y en la figura 32 se muestra la capa TOP.

Algunas de las consideraciones que se tuvieron en cuenta para realizar el diseño del circuito impreso fueron las siguientes:

- Los condensadores de desacople se ubicaron lo más cerca posible a los pines de polarización de cada uno de los componentes sobre la misma capa y con los valores sugeridos en las hojas de especificaciones de cada uno de ellos.
- La ubicación de los puertos, conectores y resistencias se realizó de tal modo que se evitaran caminos largos y así optimizar el desempeño del circuito.
- El regulador de 3.3 V se ubicó cerca a la fuente y también cerca a los componentes que deben ir polarizados con este valor de tensión.
- Se hizo un plano de tierra para cada una de las capas conectados a la misma señal (Tierra) y el aislamiento entre el plano y las rutas es fue de 0.6 mm.
- El tamaño promedio de las rutas fue de 0.6 mm a excepción de aquellas rutas cercanas a los pines de los amplificadores operacionales, ya que el tamaño entre pines no permitía un tamaño de ruta mayor a 0.37 mm.

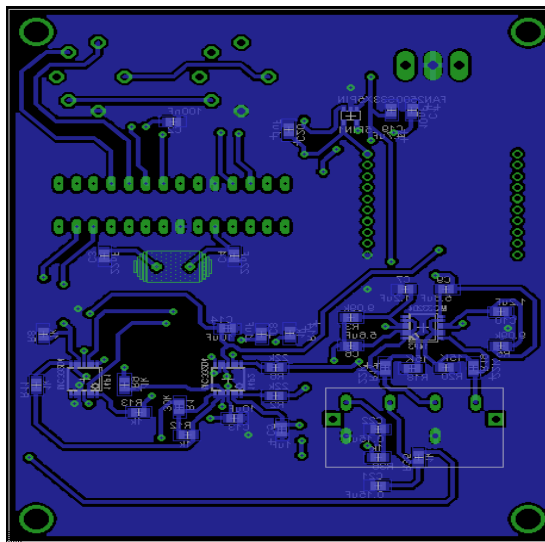


Figura 31. Capa inferior del circuito impreso implementado.

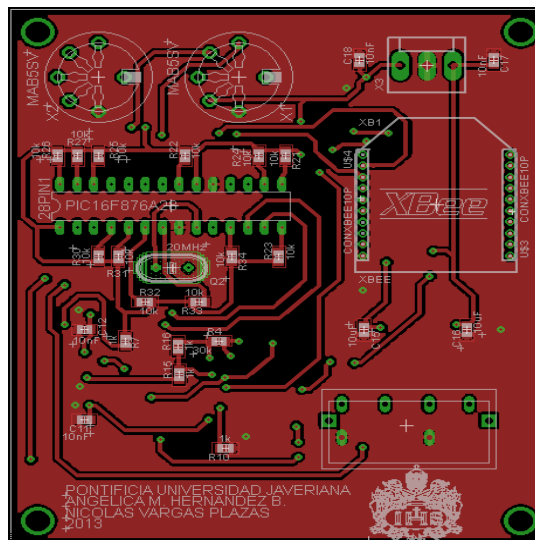


Figura 32. Capa superior del circuito impreso implementado.

Para observar el esquemático completo del circuito y el circuito impreso con los componentes soldados, referirse al anexo 6.

## 8. ARQUITECTURA PARA LA CODIFICACIÓN DE DATOS EN SIMULINK

Como se dijo anteriormente, por medio del bloque Stream server de QuaRC® es posible establecer una comunicación inalámbrica entre la tarjeta de adquisición de datos y Simulink, de tal modo que se puede realizar el control de la planta en tiempo real. Los parámetros de configuración que se establecieron para la comunicación mediante éste bloque de QuaRC® son los siguientes:

- Protocolo de comunicación: Serial
- Velocidad de transmisión: 38400 baud
- Bit de paridad: Ninguno
- Bits de parada: 1
- Tamaño de la palabra: 8 bits
- Frecuencia de muestreo: 1 kHz
- Control de transmisión: Maximum Throughput
- Tamaño en muestras de cola FIFO de recepción y transmisión: 400

Estas características fueron elegidas con los siguientes criterios:

- La conexión del módulo XBee con el computador se realiza por medio de la tarjeta XBee Explorer USB, por lo tanto el ordenador reconoce éste dispositivo como un puerto serial virtual, de tal modo que la comunicación inalámbrica se emula en el computador como un protocolo de puerto serial.
- La velocidad de transmisión de 38400 baud fue seleccionada, debido a que cumple con el ancho de banda requerido por el helicóptero para su controlabilidad, el cual es de 1.85 Hz, como puede detallarse en el anexo 2. Por otro lado, la implementación por máquina de estados de la toma de datos de los encoders por medio del microcontrolador aunque no obtiene resultados tan favorables como con 115200 baudios, como se detallará en la sección de pruebas, permite estabilizar al helicóptero mediante el uso de varios controladores sin perder muchos datos de conteos de encoders, como no ocurre con 115200 baudios, que no logra estabilizar al helicóptero con alguno de los controladores implementados. Esto último también será detallado en la sección 9 y 10 del documento.

- Dado que la comunicación vía serial UART entre el microcontrolador, los módulos XBee y el bloque stream server de QuaRC® es asíncrona como se explicó en la sección 6.2.1.1.1, se debe indicar en cada trama que se envía, cuando empieza un dato y cuando termina. Es por esto, que se adiciona un bit de parada en la transmisión, de tal modo que se pueda identificar cada dato correspondiente a las señales de los encoders y a las señales de voltaje que se envían a las UPM.
- Como se dijo anteriormente, mediante el uso de Maximum Throughput se maximiza la utilización del ancho de banda de comunicación, por lo que al utilizar una velocidad alta como 38400 baudios, los paquetes enviados y recibidos pueden estar incompletos si se usan latencias mínimas al tratar de vaciar la secuencia subyacente en cada instante de muestreo, lo que hace pertinente usar esta opción con Maximum Throughput que asegura que los paquetes lleguen completos antes de ser enviados.
- El tamaño de la cola FIFO, que corresponde al tamaño con el cual se llena este buffer antes de transmitir o recibir datos como se explicó anteriormente, se escogió de 400 muestras, puesto que al realizar pruebas con tamaños superiores, la respuesta de los datos recibidos tiene un tiempo de retardo muy grande desde que el helicóptero se mueve hasta que Simulink procesa los datos, lo cual se debe a que el buffer espera muchas muestras para poder desocuparse y enviar esos datos, por lo que en ese tiempo de espera, muchos datos ya se han acumulado o perdido. Este comportamiento de tiempos de espera es usualmente modelado como un problema de teoría de colas, siendo  $\lambda$  la tasa de llegada de cada muestra al buffer de recepción del bloque stream server y  $\mu$  la tasa promedio en la que cada muestra será desocupada del buffer:

$$T_{espera} = \frac{\lambda}{\mu(\mu - \lambda)}$$

Por otro lado, con un tamaño menor, existe una pérdida de datos que se debe básicamente a que cuando el buffer está listo para desocupar los datos de la cola, muchos datos no han llegado a almacenarse en él, por lo que en ese instante de tiempo no hay recepción de datos suficiente para ser procesados en el sistema.

Cuando el bloque stream server captura los datos de la tarjeta, que corresponde a la señal multiplexada de los conteos de los pulsos de los encoders de pitch y yaw, se realiza un algoritmo con ayuda de bloques de Simulink y código fuente de MATLAB® para reconstruir de forma correcta estos datos y enviarlos al lazo de realimentación.

Primero los datos que se reciben constantemente pasan por un bloque de Simulink llamado Buffer, el cual se encarga de redistribuir los datos en vectores de un tamaño constante, en este caso el buffer toma 8 muestras, que corresponden a la longitud de los datos que se envían desde el microcontrolador. Esto implica que la tasa de muestreo a la cual se ejecutan los datos de ahí en adelante ya no será de 1 kHz sino de 125 Hz, lo cual genera cambios en el sistema de control, como se enunciarán en la sección de análisis de resultados. Este bloque se utiliza para poder implementar un algoritmo en MATLAB® que sea capaz de demultiplexar y reconstruir los datos que el microcontrolador ha dividido en paquetes de 8 bits, puesto que el código implementado requiere un vector de tamaño constante para ejecutar su acción.

Por medio del bloque MATLAB® Function que posee Simulink, es posible implementar un algoritmo de MATLAB® y ejecutarlo en tiempo real con todo el sistema de QuaRC®. Este algoritmo mediante un ciclo evalúa constantemente si el valor que llega proveniente del buffer corresponde a la bandera que se ha asignado en el microcontrolador para la identificación de las tramas.

Cuando este lo identifica, etiqueta las muestras siguientes como el primer y segundo byte de pitch y yaw respectivamente, ya que la comunicación que se establece asegura que los datos lleguen en el mismo orden en el cual se enviaron desde la tarjeta de adquisición, como se dijo en secciones anteriores.



Con estos datos identificados, se concatenan los dos bytes de pitch en un solo vector, al igual que los dos de yaw. Por último, cada uno de estos vectores se dirige a una salida distinta del bloque MATLAB® Function (figura 33), por lo que han sido demultiplexados y el lazo de control ya puede trabajar con estos dos datos de manera independiente. En el anexo 7 se puede observar el código fuente implementado para la concatenación y demultiplexación de los datos provenientes de los encoders.

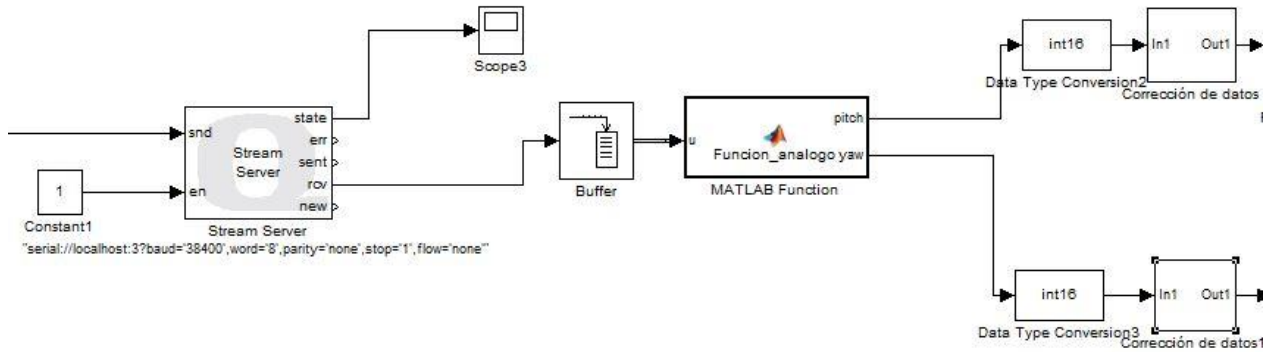


Figura 33. Bloques de codificación de datos de encoders

A continuación se describirá este algoritmo mediante un diagrama de flujo:

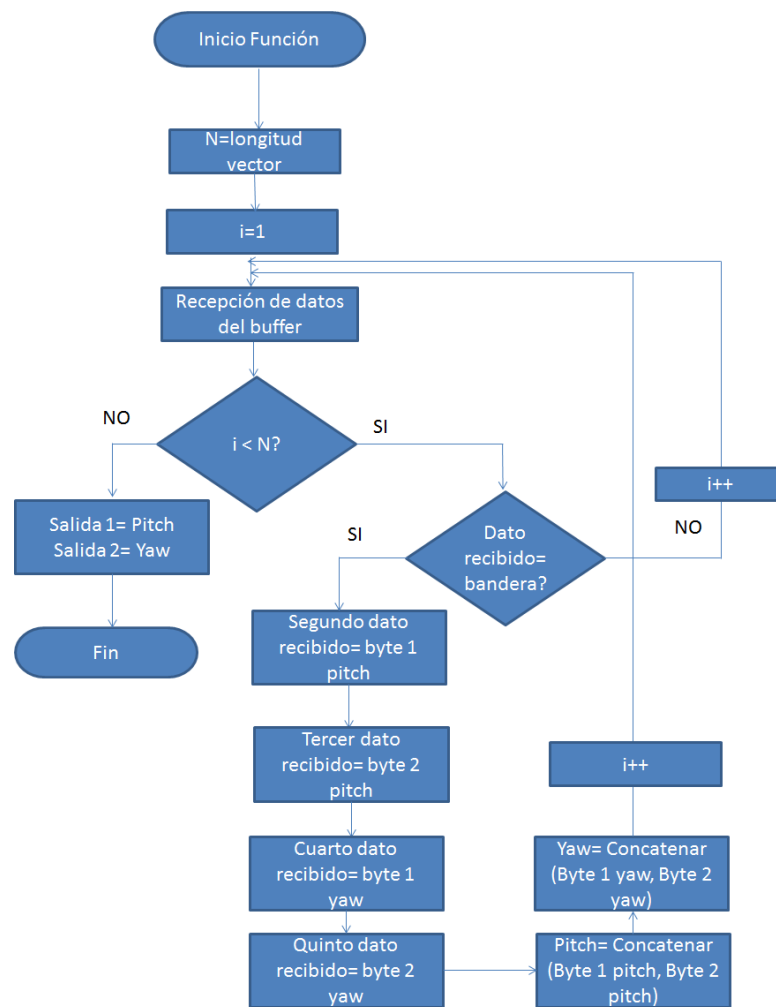


Figura 34. Diagrama de flujo del algoritmo en MATLAB®

Las señales obtenidas al concatenar y demultiplexar las señales provenientes de los encoders del helicóptero mediante la función, están compuestas cada una por ocho señales que corresponden al vector que ha arrojado el algoritmo. Cada una de estas ocho señales contiene la información de los conteos de los encoders en distintos intervalos de tiempo.

Para lograr acoplar las señales de pitch y yaw al sistema de control, es necesario unir estas señales en una sola que sea unidimensional. Esto se realizó mediante la implementación de una función y de bloques de Simulink que multiplexan, promedian estas señales y eliminan valores nulos de cada una de estas, que no pertenecen a la señal original. En el anexo 8 se puede detallar el código implementado.

Posterior a ello, la señal de salida presenta valores indefinidos que se generan del algoritmo anterior, además la comunicación genera ruido en las señales recibidas de los encoders, lo que da como resultado datos no coherentes como valores por encima de la máxima resolución de los encoders o valores que no corresponden al movimiento que se está generando en el helicóptero. Para evitar esto, se implementó una función que elimina estos valores no coherentes a partir de las muestras anteriormente obtenidas. Para realizar el algoritmo, se necesitan almacenar datos en un buffer para tomar las muestras en cada un determinado instante de tiempo. En este caso se almacenan cuatro muestras en el buffer para tener información sobre datos anteriores a cada muestra.

Esto implica que la frecuencia de muestreo del sistema en este punto se reduce a 31.25 Hz (Frecuencia de muestreo / número de muestras almacenadas en buffer), lo que conlleva a varios cambios en el lazo de control como se detallará en la sección de pruebas y análisis de resultados. En el anexo 14 se puede observar el código del algoritmo debidamente documentado.

## **9. PROTOCOLO DE PRUEBAS**

Las pruebas que se realizaron sobre el sistema de adquisición de datos se basaron primero en la comprobación de las señales y niveles de voltaje requeridos en la tarjeta de adquisición, al enviar y recibir señales por medio del módulo XBee. Posterior a ello, se verificó la recepción de los datos en Simulink mediante el bloque de comunicación de QuaRC® y las señales obtenidas a partir del procesamiento de codificación de datos explicado anteriormente. Por último se realizaron pruebas de la tarjeta ante entradas paso en lazo abierto y con varios controladores en lazo cerrado ante distintas condiciones en la comunicación que serán descritas en la sección 9.4.

### **9.1 PRUEBAS EN TARJETA DE ADQUISICION**

En esta sección se mostrarán los resultados y se explicará cada una de los casos que se tuvo en cuenta y la razón por la cual estas pruebas fueron pertinentes.

La codificación que se realizó en Simulink divide los datos que se van a enviar en valores desde 0 hasta 255, correspondientes al porcentaje de ciclo útil que se tenga en la salida del PWM, pero esta forma de codificación no es conveniente para la aplicación que se está realizando, por eso se utilizó un factor de escala para convertir esos valores de 0 a 255 a sus correspondientes valores de voltaje entre -5 V y 5 V, en la figura 35 se observa la relación entre los datos de ciclo útil y sus correspondientes valores en voltajes. El factor de escala se puede determinar por una relación de característica lineal de esta forma:

$$y = 0,0388x - 4,95$$

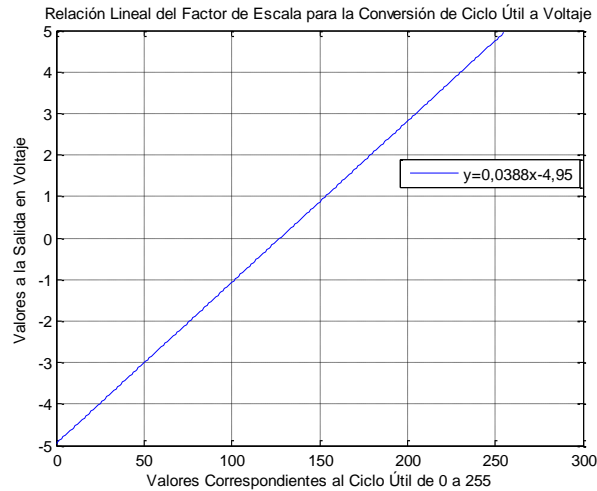


Figura 35. Conversión entre los datos de ciclo útil a su voltaje correspondiente

A continuación se muestra la arquitectura de Simulink que se utilizó para verificar el correcto funcionamiento del envío de un valor al microcontrolador para generar una señal de PWM correspondiente al valor enviado. Los subsistemas de conversión 1 y 2 (Ver figura 37) contienen el factor de conversión del valor de voltaje a la codificación de datos, como se puede detallar en el anexo 9.

En estas pruebas se realizó la comunicación a una tasa de 38400 baudios, que corresponde a la velocidad final implementada para el sistema inalámbrico.

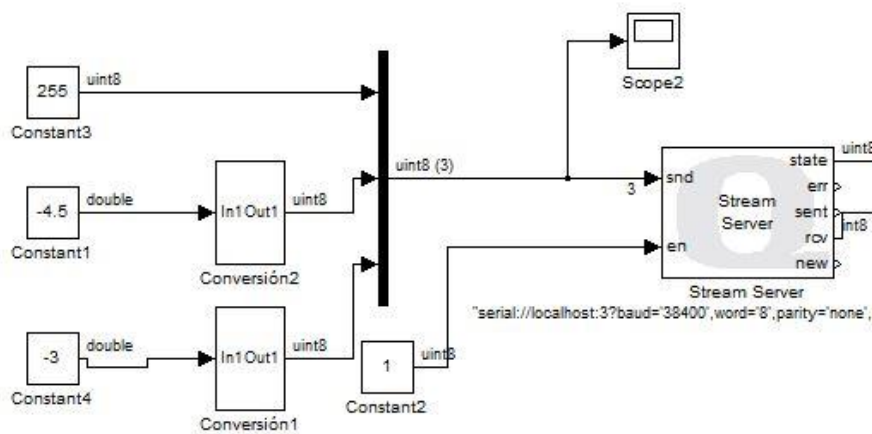


Figura 36. Bloques en Simulink que permite enviar los valores de voltaje para generar las señales de PWM

En la siguiente figura se puede observar la señal de PWM proveniente del microcontrolador al enviar un valor correspondiente a 0 V desde Simulink. La señal se tiene ciclo útil del 50% por lo anteriormente explicado, es decir que en la codificación de los datos en realidad se está enviando el valor de 127 hacia el microcontrolador. Además la señal tiene una amplitud de aproximadamente 3.3 V como era de esperarse al polarizar el microcontrolador con este voltaje.

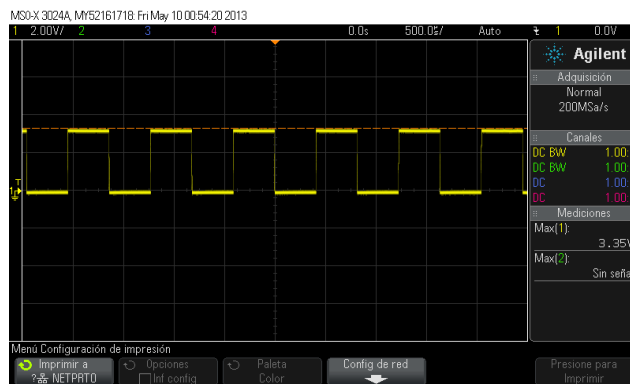


Figura 37. Salida de señal PWM en el microcontrolador al enviar el valor 0 V desde Simulink.

En la figura 38 se puede observar otro ejemplo de la señal de PWM desde el microcontrolador, en este caso se tiene un valor de 2 V enviado desde Simulink lo cual corresponde a un valor de 179 con la codificación de datos realizada.

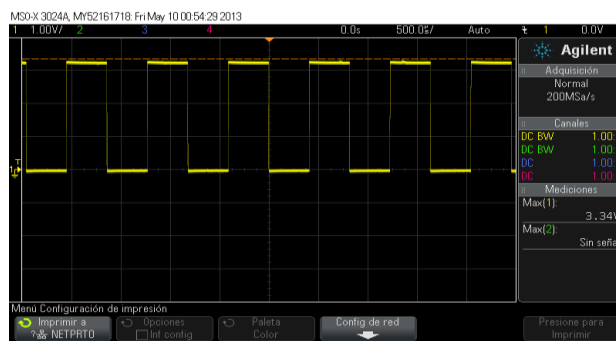


Figura 38. Salida de señal PWM en el microcontrolador al enviar el valor 2 V desde Simulink

En la figura 39 se puede ver la salida del PWM con un ciclo útil del 20% aproximadamente, esto se presenta cuando se envía el valor -3 V desde Simulink, y su valor correspondiente en la codificación de datos sería de aproximadamente 50.2.

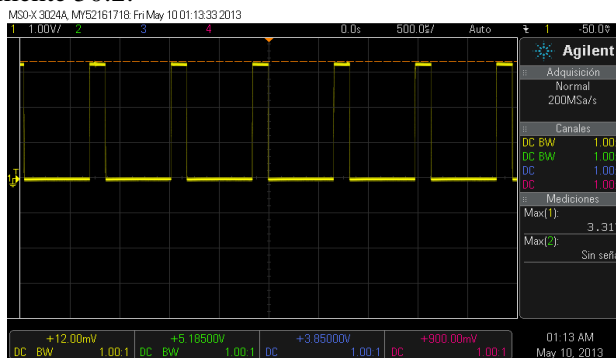


Figura 39. Salida de señal PWM en el microcontrolador al enviar el valor -3 V desde Simulink

En la figura que se muestra a continuación se ve como el microcontrolador puede demultiplexar cada una de las señales respectivas a Pitch y a Yaw enviadas a través del canal inalámbrico. La señal que está en amarillo pertenece a Pitch con -4.5 V enviados desde Simulink y la azul pertenece a Yaw con 2 V, se observa que los ciclos útiles son 4.5% y 70.2% respectivamente. Además a estos porcentajes corresponden los valores de 11.6 y 179 en la codificación de datos de Simulink.

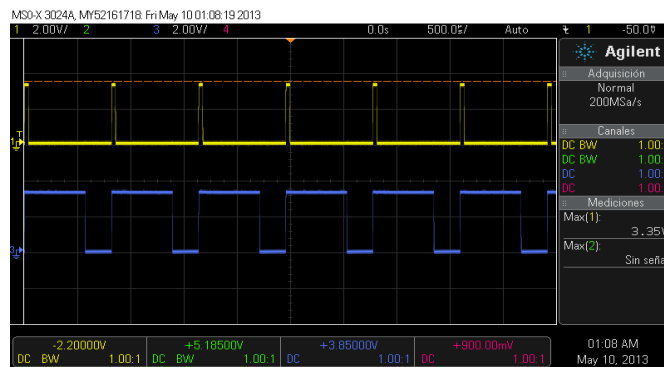


Figura 40. Señales de PWM en el microcontrolador, Pitch en -4.5 V (amarillo) y Yaw en 2 V (azul)

La figura 41 muestra una señal de PWM con ciclo útil del 80.3%, esta corresponde a un valor de 3 V enviado desde Simulink ó una codificación de 204.7, como se puede observar la señal se encuentra entre 0 y 3.3V de amplitud, pero para que pueda ser enviada a la UPM la señal debería estar entre -5V y +5V para corregir esto se utilizó un circuito restador, cuya arquitectura se mostró anteriormente en el desarrollo teórico en las secciones 6.2.5 y 6.2.6.

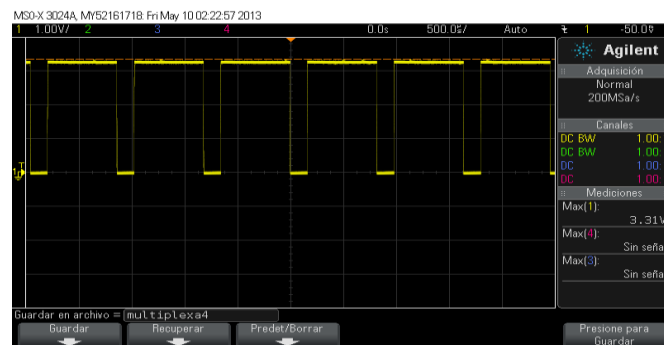


Figura 41. Señal de PWM en el microcontrolador al enviar el valor 3 V desde Simulink

La corrección de nivel de la señal de PWM se realizó con el circuito restador de la sección 6.2.5, el cual acondiciona el nivel de la señal tal que ésta quede con la misma amplitud de 3.3 Vpp pero centrada en cero, es decir que su amplitud será de  $\pm 1.65$  V aproximadamente como se observa en la figura 42.

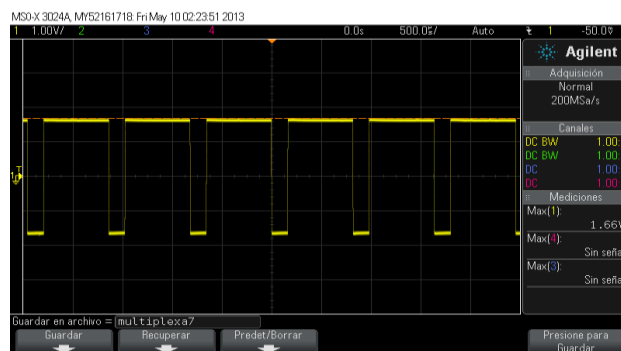


Figura 42. Señal de PWM a la salida del circuito restador que adecua la señal entre  $\pm 1.65$  V

En la figura 43 se observa la señal después de los filtros como un DC correspondiente al valor del ciclo útil como lo explica la sección 6.2.4 y adicionalmente esta multiplicada por la ganancia de 3.0 de la sección 6.2.6. Con estas correcciones la señal ya queda acondicionada para ser enviada a través de los conectores RCA hacia las UPM que permiten hacer el control sobre los motores del helicóptero 2DOF de Quanser®. En la figura se observa que el valor DC obtenido es 3 V aproximadamente, que fue el valor enviado desde Simulink.

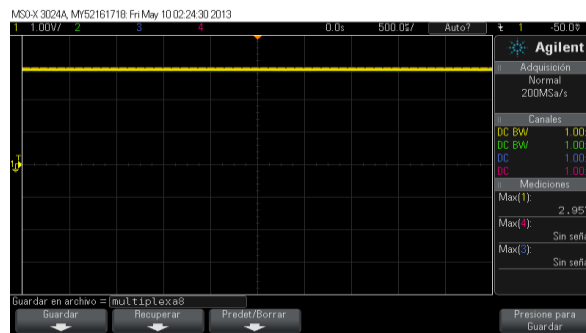


Figura 43. Señal de control hacia la UPM de 3 V a la salida del filtro pasa bajo con ganancia de 2.9.

En la siguiente tabla se pueden observar las pruebas que se realizaron sobre distintos valores de voltaje enviados a la tarjeta de adquisición, y su correspondiente valor obtenido a la salida de esta:

Tabla 10. Comparación de valores

SEÑAL DE CONTROL	VOLTAJE DE ALIMENTACION DEL HELICPOTERO
0 V	0 V
0,5 V	0,48 V
1 V	0,96 V
1,5 V	1,45 V
2 V	2,01 V
2,5 V	2,49 V
3 V	2,97 V
3,5 V	3,54 V
4 V	4,02 V
4,5 V	4,5 V
5 V	4,9 V

En la sección 10.1 se mostrarán gráficas ilustrando el comportamiento de estas pruebas.

## 9.2 PRUEBAS DE RECEPCION DE DATOS EN SIMULINK

A continuación se podrán observar las señales tomadas de los encoders por medio del microcontrolador. En el código de implementación en el microcontrolador se codificaron los datos de tal manera que hay una bandera con el valor escogido arbitrariamente de -120 que dará inicio a la secuencia anteriormente explicada por el algoritmo de la sección 8. La figura 44 mostrará que efectivamente los datos que llegan a la salida de recepción (RCV) del bloque stream server de QuaRC® en Simulink sigue el algoritmo propuesto y permite identificar 4 bytes, 2 pertenecientes a Pitch y 2 a Yaw. En la figura 44 se observa la bandera de -120 y enseguida el primer byte de Pitch, luego el segundo byte de Pitch seguidos por el primer y segundo byte de yaw respectivamente. Después que se termina este ciclo entonces nuevamente se ve la bandera de -120 y así sucesivamente se van tomando todos los datos de los conteos de los encoders.

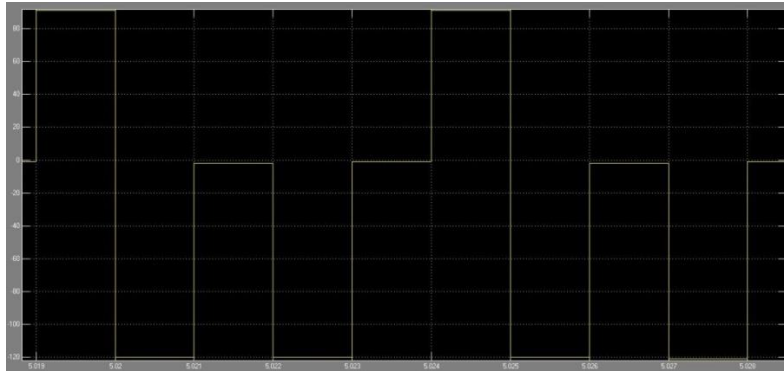


Figura 44. Salida de recepción del bloque stream server de QuaRC® mostrando los conteos de los encoders.

La figura 45 muestra la señal de Pitch concatenada después de pasar por el bloque de función de Simulink donde se implementó una función que permite tomar los 2 bytes de Pitch y los 2 de Yaw y concatenarlos cada uno por aparte para poder trabajar con esas dos señales de forma independiente de ahí en adelante. La figura 46 muestra la señal concatenada de Yaw.

Las señales que se ven en estas figuras corresponden a los conteos realizados por el microcontrolador de los encoders al mover el helicóptero en sus dos ejes Pitch y Yaw de forma aleatoria para demostrar que el algoritmo de concatenación de las señales funciona correctamente.

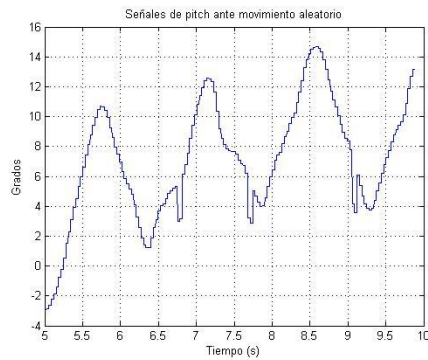


Figura 45. Señal de Pitch concatenada

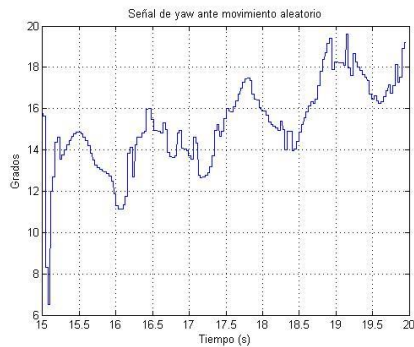


Figura 46. Señal de Yaw concatenada

### 9.3 PRUEBAS SOBRE EL SISTEMA DE CONTEO DE PULSOS DE LOS ENCODERS

Uno de los parámetros más importantes a tener en cuenta al realizar el control sobre el helicóptero, es la fiabilidad de los datos que recibe Simulink de acuerdo a los conteos de los encoders de este que determinan los grados que se ha movido cada uno de los motores al recibir las señales de control

respectivas. Por tanto al perder datos correspondientes a los conteos de alguno de los encoders, no se logra observar el resultado correcto del funcionamiento del sistema por ejemplo en lazo abierto, pero en lazo cerrado no sólo se pierde resolución de las medidas, sino al realimentar el sistema con estas señales, el control no se realizará correctamente puesto que la referencia que tiene el helicóptero podría ser otra si no se obtienen bien estos datos.

El resultado exitoso de la obtención de estos conteos depende de varios factores. El primero, es la implementación realizada en el microcontrolador, puesto que al poder optimizar código innecesario, se puede lograr una mayor rapidez en la generación de estos datos de acuerdo a la velocidad de transmisión definida. Por otro lado, la velocidad de transmisión es un hecho determinante en el resultado de estos conteos, puesto que a velocidades muy bajas, la implementación realizada en el microcontrolador no es capaz de contar y enviar todos los datos de los encoders debido su alta resolución. Por último, la configuración de los buffers de recepción y envío del bloque de comunicación de QuaRC® también es un factor muy importante para lograr un buen resultado de los datos recibidos, puesto que al acumular muchos bytes en el buffer, se retarda el envío de los datos al esperar que se llene ese buffer y por tanto puede que estas señales no se actualicen debidamente, por lo que se pueden llegar a perder muchos datos importantes, pero si el buffer no se llena lo suficiente de acuerdo a los datos que envía el microcontrolador, el algoritmo de demultiplexación y concatenación no realizará el procedimiento asignado

En la tabla 11 se muestra el rendimiento del sistema de conteo de los encoders realizando las pruebas sobre los grados que se obtienen de pitch en lazo de abierto en relación con el cambio del buffer de envío del bloque stream server de QuaRC®, a las tres velocidades de transmisión más altas soportadas por los módulos XBee, teniendo en cuenta que el tamaño mínimo de buffer de envío con el que se transmiten los datos es configurado a 70 bytes, puesto que a una menor tasa, se pierde la mayor parte de los datos y no es posible obtener resultados coherentes. El porcentaje de rendimiento se obtiene a partir de la comparación de los grados obtenidos por la tarjeta Q8-USB de Quanser® al someter a un movimiento aleatorio a pitch respecto a cuantos grados se han perdido por el sistema de conteo implementado en la comunicación inalámbrica.

$$\text{Porcentaje de rendimiento} = \frac{(\text{Grados medidos en tarjeta de adquisición})}{\text{Grados medidos en Q8 - USB}} \times 100\%$$

Tabla 11. Pruebas del rendimiento del sistema de conteo de los Encoders a diferentes velocidades

<b>Rendimiento de los Encoders</b>			
	<b>Velocidad 115.200 Baudios</b>	<b>Velocidad 57.600 Baudios</b>	<b>Velocidad 38.400 Baudios</b>
<b>Buffer Send (Bytes)</b>	<b>Rendimiento</b>	<b>Rendimiento</b>	<b>Rendimiento</b>
<b>70</b>	98.82%	91.76%	82.35%
<b>100</b>	99.3%	94.11%	76.47%
<b>130</b>	99.3%	64.71%	71.76%
<b>160</b>	98.82%	47.06%	35.29%
<b>190</b>	99.3%	23.53%	29.41%
<b>220</b>	99.3%	22.35%	21.17%
<b>150</b>	99.3%	18.82%	23.53%
<b>280</b>	99.3%	16.47%	23.53%
<b>310</b>	99.3%	89.41%	47.06%
<b>350</b>	99.3%	94.11%	58.82%
<b>400</b>	98.82%	90.5%	70.59%



<b>450</b>	95.29%	88.2%	58.82%
<b>500</b>	95.29%	76.4%	35.29%
<b>550</b>	96.47%	61.2%	29.41%
<b>600</b>	98.81%	57.6%	47.06%

#### 9.4 PRUEBAS DEL SISTEMA EN LAZO ABIERTO

Las pruebas que se realizaron del sistema inalámbrico en Simulink tiene como base el sistema Demo que incluye Quanser® para el control del helicóptero 2-DOF, reemplazando los bloques de conexión con las tarjetas de Quanser®, por el sistema de comunicación inalámbrica mediante el bloque stream server de QuaRC® y la arquitectura de codificación de los datos provenientes del helicóptero. Esto se realizó con el fin de observar el comportamiento adecuado de la tarjeta de adquisición ante cierta entrada de voltaje de excitación de los motores.

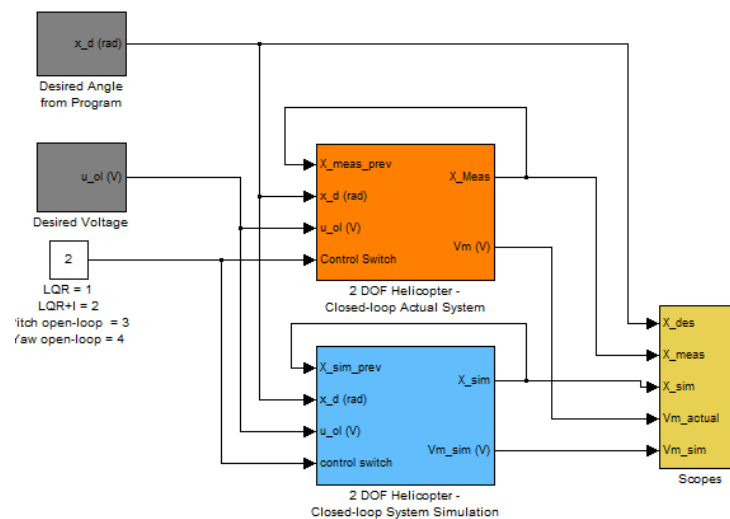


Figura 47. Sistema de simulación y control de Quanser® para helicóptero 2-DOF

Este sistema demo de Quanser® se compone de las siguientes partes:

- Bloque de lazo abierto para pitch
- Bloque de lazo abierto para yaw
- Bloque de lazo cerrado con controlador LQR
- Bloque de lazo cerrado con controlador LQI
- Bloque de conexión con sistema de comunicación, con bloques saturadores de voltaje, ganancia para conversión de conteos a grados y filtros derivativos para obtención de estados de velocidad de pitch y yaw.
- Bloque de simulación con modelo no lineal de la planta.

Para realizar las pruebas de lazo abierto, se utilizó la configuración lazo abierto-pitch que muestran el comportamiento de los dos ejes del helicóptero al aplicar un voltaje de entrada solamente para pitch sin variar la entrada de yaw.

Aplicando como entrada al sistema una señal cuadrada de amplitud 2 V en pitch y 0 V en yaw con frecuencia de 0.1 Hz, se obtienen los siguientes resultados para pitch y yaw:

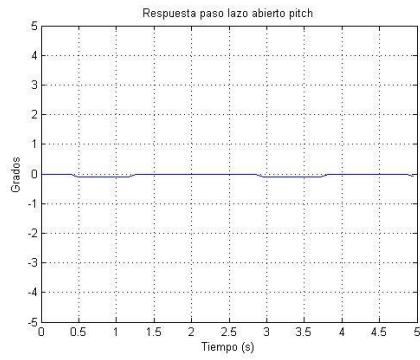


Figura 48. Respuesta de pitch lazo abierto con 2 V de amplitud

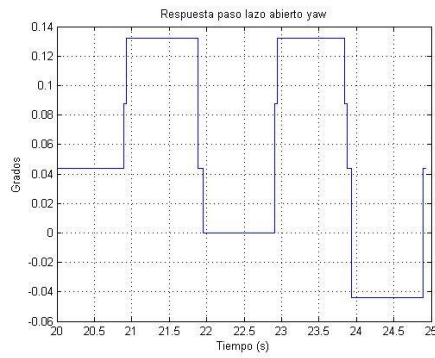


Figura 49. Respuesta de yaw lazo abierto con 2 V de amplitud

Ahora, aplicando la misma señal de entrada pero con 3 V de amplitud:

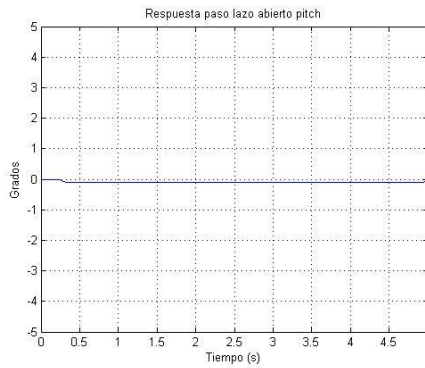


Figura 50. Respuesta de pitch lazo abierto con 3 V de amplitud

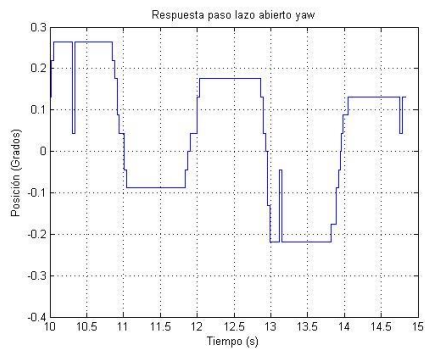


Figura 51. Respuesta de yaw lazo abierto con 3 V de amplitud

En la sección 10 de análisis de resultados se presentará una comparación entre los resultados obtenidos con la tarjeta de adquisición inalámbrica y la tarjeta Q8-USB de Quanser® para lazo abierto, el análisis correspondiente a estos datos y las conclusiones obtenidas a partir de estos.

## 9.5 PRUEBAS DEL SISTEMA EN LAZO CERRADO TARJETA INALAMBRICA-Q8 USB

El sistema de control en lazo cerrado mediante la tarjeta de adquisición inalámbrica tiene dos problemas a resolver. El primero de ellos fue establecer los parámetros correctos de la comunicación, esto incluye la velocidad de transmisión y los parámetros del bloque de comunicación con Quanser® para garantizar la correcta transmisión y recepción de los datos y a la velocidad suficiente para que el control sea instantáneo. Por otro lado, se debe asegurar que los datos provenientes de los encoders no sufran pérdidas como se detalló anteriormente que ante determinadas condiciones estos datos tenían un buen o mal comportamiento.

Para lograr identificar los posibles errores de cada uno de estos problemas y poder corregirlos de forma separada, el primero fue observar el rendimiento del algoritmo de conteo de los encoders como se mostró anteriormente y el segundo fue hacer una conexión alámbrica-inalámbrica para evaluar distintos controladores ante diversas situaciones. Esta conexión se realizó tomando los datos de los encoders y llevándolos al lazo de realimentación por medio de la tarjeta Q8-USB de Quanser® y la recepción de datos y envío de señales de control a los módulos de potencia por medio de la tarjeta de adquisición de datos inalámbrica. Esto se realizó con el fin de evitar en un principio pérdida de datos por el algoritmo de conteo de los encoders y poder evaluar cada controlador y a partir de esto, realizar las pruebas finales con todo el sistema de conexión inalámbrico.

Después de realizar esta conexión se probaron y evaluaron 5 controladores distintos en el sistema:

- Controlador LQR continuo
- Controlador LQI (LQR + Integrador) continuo
- Controlador proporcional por realimentación de estados discreto 1
- Controlador proporcional por realimentación de estados discreto 2
- Controlador proporcional por realimentación de estados continuo

Los dos primeros controladores están diseñados e implementados en el sistema demo de Quanser®, por lo que su funcionamiento correcto ya está establecido. Por favor referirse al anexo 10 para detallar como se definieron los parámetros de cada uno.

Los controladores discretos fueron diseñados tomando como base de los polos en lazo cerrado del controlador LQR, pero mediante el diseño de controlador proporcional por realimentación de estados a través de la función place que posee MATLAB®. Básicamente, las matrices A, B, C, D fueron discretizadas con una frecuencia de muestreo de 1 kHz y posterior a esto los polos en lazo cerrado fueron discretizados a ésta misma frecuencia de muestreo, que corresponde a la frecuencia de muestreo original del sistema inalámbrico:

$$P(z) = e^{-s \cdot T_s}$$

El segundo controlador discreto tiene las mismas características pero con la diferencia que fue discretizado a una frecuencia de muestreo de 31.25 Hz. Esto se realizó para observar los posibles cambios que pueda tener el sistema ante variaciones en el muestreo de las señales, teniendo en cuenta que por la implementación realizada en la codificación de datos en Simulink para la toma de las señales de los encoders (Referirse a la sección 8), la frecuencia de muestreo se reduce a 31.25 Hz.

El último controlador fue diseñado al igual que los controladores discretos, mediante la técnica de realimentación de estados a partir de los polos de lazo cerrado obtenidos por el control LQR, de tal forma que se pudiera tener una comparación frente estas dos leyes de control en el sistema. Para observar el diseño detallado de estos tres últimos controladores por favor referirse al anexo 11.

Para evaluar el desempeño de estos controladores, se realizaron pruebas variando la velocidad de transmisión de los módulos XBee (38400 baudios, 57600 baudios, 115200 baudios) y variando el tamaño del buffer de transmisión del bloque stream server de Quanser®, que como se podrá observar, es un factor determinante para la estabilización del helicóptero. Se utilizaron señales de entrada con frecuencia de 0.1 Hz y se evaluó la estabilidad para 10° en pitch y 30° en yaw. A continuación se muestran los resultados obtenidos para cada caso de los parámetros ya descritos y con cada uno de los cinco controladores implementados:

Tabla 12. Desempeño de los controladores con una velocidad de transmisión de 38.400 Baudios.

<b>Velocidad 38400</b>										
<b>Cont rol/B uffer</b>	<b>60</b>	<b>100</b>	<b>140</b>	<b>186</b>	<b>220</b>	<b>260</b>	<b>300</b>	<b>340</b>	<b>380</b>	<b>420</b>
<b>LQR</b>	No Logró Estabilizarse	No Logró Estabilizarse	Logró Estabilizarse con oscilaciones	Logró Estabilizarse	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse
<b>LQI</b>	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kd1</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	Logró Estabilizarse	Logró Estabilizarse con oscilaciones	Logró Estabilizarse con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kd2</b>	Logró Estabilizarse con oscilaciones	Logró Estabilizarse	Logró Estabilizarse con oscilaciones	Logró Estabilizarse	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kc</b>	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones	Logró Estabilizarse con oscilaciones	Logró Estabilizarse con oscilaciones	Logró Estabilizarse con oscilaciones	Logró Estabilizarse con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse

Tabla 13. Desempeño de los controladores con una velocidad de transmisión de 57.600 Baudios.

<b>Velocidad 57600</b>										
<b>Contr ol/Buf fer</b>	<b>60</b>	<b>100</b>	<b>140</b>	<b>186</b>	<b>220</b>	<b>260</b>	<b>300</b>	<b>340</b>	<b>380</b>	<b>420</b>
<b>LQR</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	Logró Estabilizarse e con oscilaciones	Logró Estabilizarse e con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	Logró Estabilizarse con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>LQI</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	Logró Estabilizarse e con pocas oscilaciones	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kd1</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	Logró Estabilizarse e con pocas oscilaciones	Logró Estabilizarse e con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kd2</b>	No Logró	No Logró	No Logró	Logró Estabilizarse	Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró	No Logró	No Logró

	Estabilizarse	Estabilizarse	Estabilizarse	e	e con oscilaciones grandes	se	se	Estabilizarse	Estabilizarse	Estabilizarse
<b>Kc</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	Logró Estabilizarse e con oscilaciones grandes	Logró Estabilizarse e con oscilaciones grandes	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse

Tabla 14. Desempeño de los controladores con una velocidad de transmisión de 115.200 Baudios.

Velocidad 115200										
Control/Buffer	60	100	140	186	220	260	300	340	380	420
<b>LQR</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>LQI</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kd1</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kd2</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse
<b>Kc</b>	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse	No Logró Estabilizarse

Como se pudo observar, el parámetro de buffer SND es determinante para la estabilización, como se mostrará con más detalle en la sección de análisis de resultados. A continuación se mostrarán las gráficas obtenidas de las pruebas exitosas del sistema de control, es decir, de las pruebas que lograron estabilizar el helicóptero 2-DOF. En la sección de análisis de resultados se podrá observar de forma gráfica y analítica estos resultados obtenidos.

Con los controladores LQI y LQR a una velocidad de 38400 baudios, 186 bytes en buffer snd y con 10 grados como referencia para pitch:

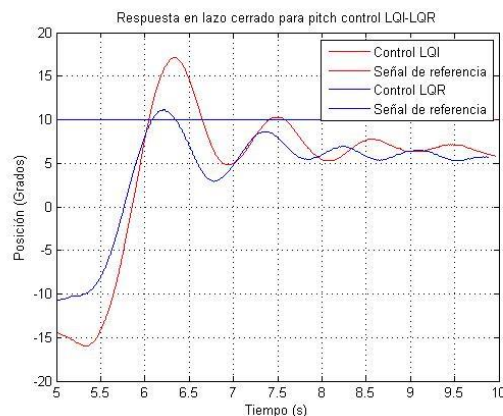


Figura 52. Respuesta de pitch lazo cerrado LQR y LQI con señal de referencia de 10 grados

En el caso de yaw, con una señal de referencia de 10 grados:



Figura 53. Respuesta de yaw lazo cerrado LQR y LQI con señal de referencia de 10 grados

Con los controladores proporcional discreto 1, proporcional discreto 2 y proporcional continuo a una velocidad de 38400 baudios, 186 bytes en buffer snd y con 10 grados como referencia para pitch:

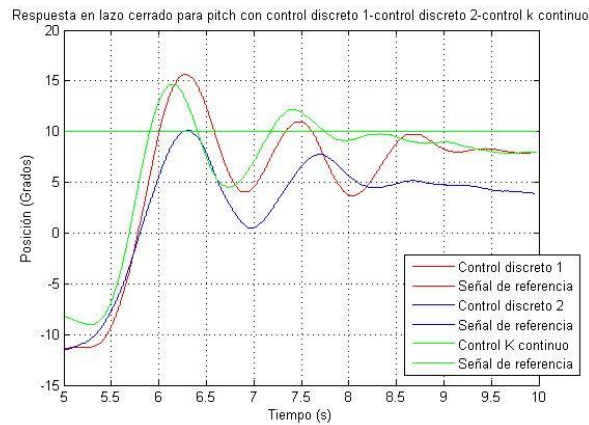


Figura 54. Respuesta de pitch lazo cerrado con control proporcional discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados

En el caso de yaw, con una señal de referencia de 10 grados:

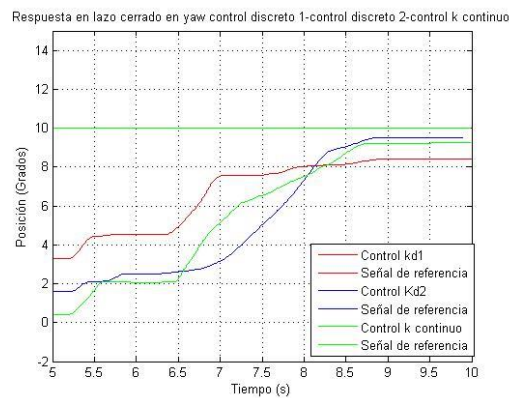


Figura 55. Respuesta de yaw lazo cerrado con control proporcional discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados

Con los controladores LQR y LQI a una velocidad de 57600 baudios, 186 bytes en buffer snd y con 10 grados como referencia para pitch:

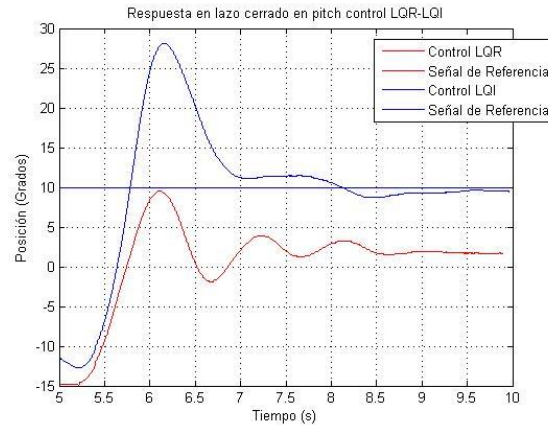


Figura 56. Respuesta de pitch lazo cerrado con LQR y LQI con señal de referencia de 10 grados

Con los controladores LQR y LQI a una velocidad de 57600 baudios, 186 bytes en buffer snd y con 10 como referencia para yaw:

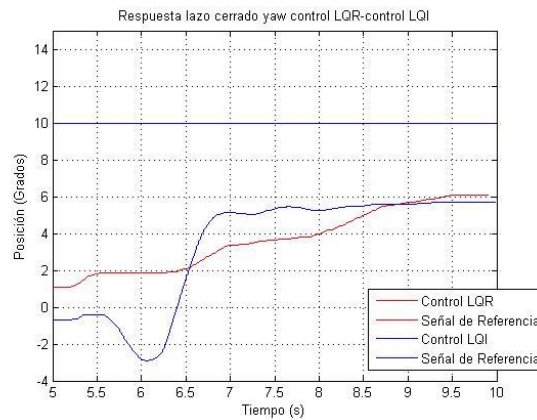


Figura 57. Respuesta de yaw lazo cerrado con LQR y LQI con señal de referencia de 10 grados

Con los controladores proporcional discreto 1, proporcional discreto 2 y proporcional continuo a una velocidad de 57600 baudios, 186 bytes en buffer snd y con 10 grados:

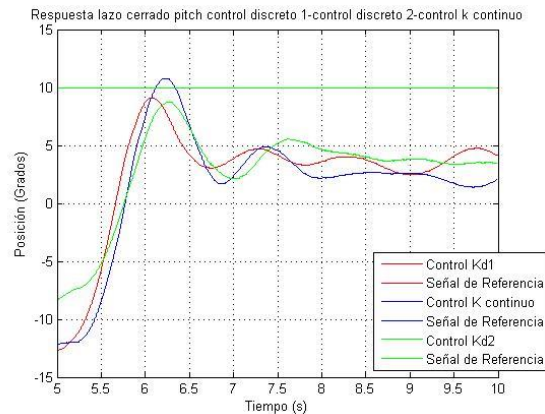


Figura 58. Respuesta de pitch lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados

En el caso de yaw, con una señal de referencia de 10 grados:

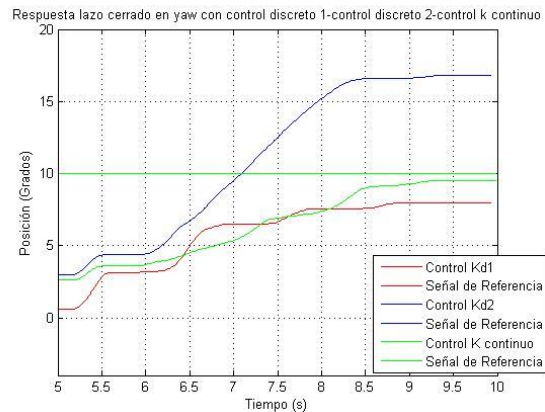


Figura 59. Respuesta de yaw lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados

En la sección 10 de análisis de resultados se realizará una comparación de estos datos obtenidos con las simulaciones realizadas solamente con la tarjeta Q8-USB de Quanser® y con todo el sistema inalámbrico.

## 9.6 PRUEBAS DEL SISTEMA EN LAZO CERRADO TARJETA INALAMBRICA

Al realizar las pruebas correspondientes a los rendimientos del sistema de conteo de los encoders y la evaluación de cada uno de los controladores sobre la conexión tarjeta inalámbrica-Q8 USB, se dedujo que debe existir un balance entre el comportamiento del uno y del otro para acoplar todo el sistema de forma inalámbrica. Esto implica que el rendimiento del sistema de conteo de los encoders debe ser lo suficientemente bueno para que haya una pérdida mínima de datos, pero a la vez se necesita que las señales de control se envíen con la rapidez necesaria para que se realice el control esperado.

Puesto que ante una velocidad de 115200 baudios no se logró estabilizar en ningún caso el helicóptero, esta fue descartada para la implementación final en la tarjeta aunque tuviera un rendimiento alto en el sistema de conteo de los encoders, como se detallará en la sección de análisis de resultados. Por tanto, la velocidad que presenta un mejor balance entre el sistema de conteo y el control del helicóptero, teniendo en cuenta los tamaños de buffer de envío y recepción es 38400 baudios.

Los controladores implementados en este caso fueron:

- Controlador LQR
- Controlador LQI (LQR + Integrador)
- Controlador proporcional por realimentación de estados discreto 2
- Controlador proporcional por realimentación de estados continuo

El controlador discreto 2 fue escogido debido a que está diseñado con un tiempo de muestreo de 0.032 s, igual que la tasa final a la que se envían las señales de los encoders al lazo de realimentación, debido a los buffers que se implementaron para demultiplexar y corregir éstas señales. Además se implementó un filtro pasabajos en el sistema de corrección de datos, con el fin de suavizar los datos que no tenían coherencia en las señales respecto a los datos provenientes de los encoders, como puede detallarse en el anexo 11.

Aplicando una señal de referencia de -10 y 10 grados en pitch y yaw a una frecuencia de 0.1 Hz con un tamaño de buffer de envío de 140 bytes con los controladores LQR y LQI se obtuvieron los siguientes resultados:



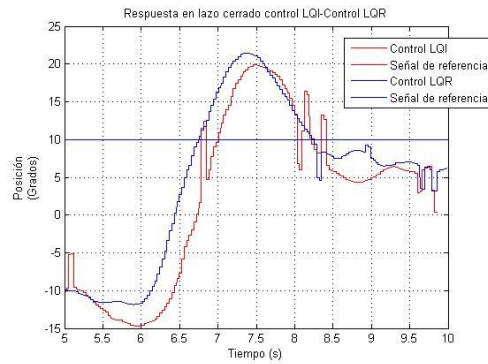


Figura 60. Respuesta de pitch lazo cerrado con controladores LQR y LQI con señal de referencia de 10 grados

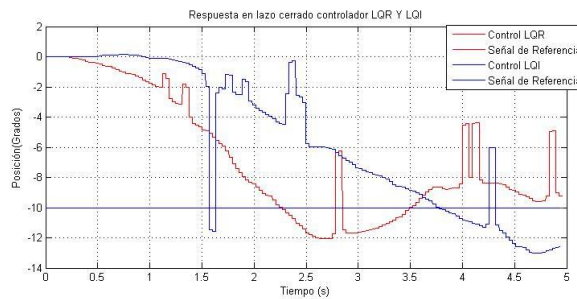


Figura 61. Respuesta de pitch lazo cerrado con controladores LQR y LQI con señal de referencia de -10 grados

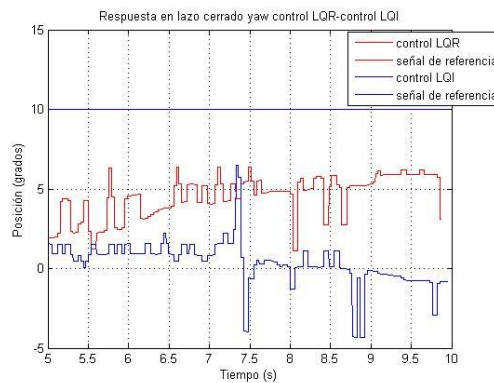


Figura 62. Respuesta de yaw lazo cerrado con controladores LQR y LQI con señal de referencia de 10 grados

Con los controladores proporcional discreto 1, proporcional discreto 2 y proporcional continuo los resultados fueron los siguientes:

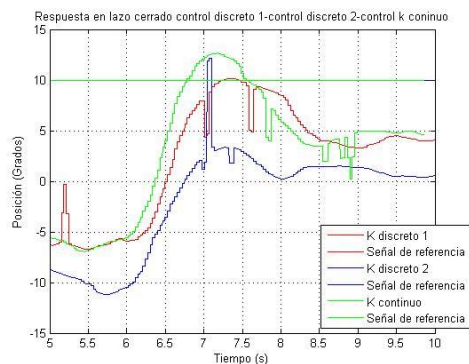


Figura 63. Respuesta de pitch lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de 10 grados

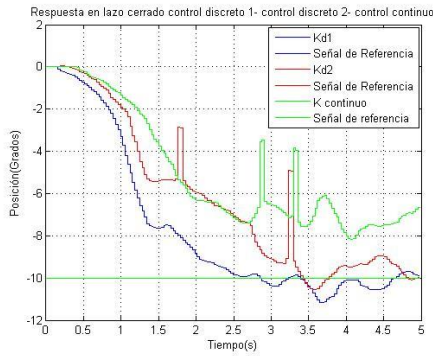


Figura 64. Respuesta de pitch lazo cerrado con controladores discreto 1, proporcional discreto 2 y proporcional continuo y señal de referencia de -10 grados

Para el caso de yaw se obtuvo lo siguiente:

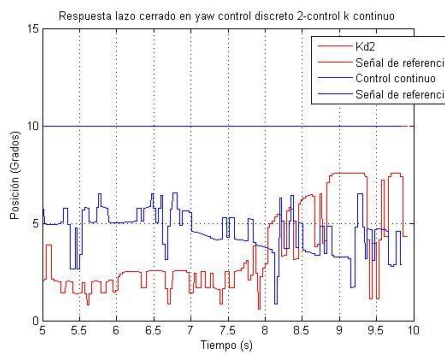


Figura 65. Respuesta de yaw lazo cerrado con controladores discreto 2 y proporcional continuo con señal de referencia de 10 grados

## 10. ANÁLISIS DE RESULTADOS

### 10.1 ERROR EN LOS VALORES DE SALIDA DEL CIRCUITO

La tabla 15 muestra la comparación entre el valor real que se está enviando desde Simulink hacia el microcontrolador en la tarjeta de adquisición y el valor que se está obteniendo a la salida de la misma después de los filtros y el ajuste de nivel necesario que se explicó anteriormente.

Tabla 15. Error en la salida hacia la UPM de la tarjeta de adquisición de datos.

Valor Simulink $y(k)$	Valor Salida $\overline{y(k)}$	$y(k) - \overline{y(k)}$
0 V	0 V	0
0,5 V	0,48 V	0,0004
1 V	0,96 V	0,0016
1,5 V	1,45 V	0,0025
2 V	2,01 V	0,0001
2,5 V	2,49 V	0,0001
3 V	2,97 V	0,0009
3,5 V	3,54 V	0,0016
4 V	4,02 V	0,0004
4,5 V	4,5 V	0
5 V	4,9 V	0,01

Haciendo un análisis de error de estos resultados mediante la siguiente ecuación para el cálculo del error porcentual:

$$EP = 100 \times \sqrt{\frac{\sum_{k=1}^n (y(k) - \overline{y(k)})^2}{\sum_{k=1}^n (y(k))^2}}$$

Al resolver la ecuación con los valores de la tabla se tiene un error porcentual del 1.35% sobre todos los datos, en la figura 67 se muestra una gráfica donde se ve la diferencia de  $y(k) - \overline{y(k)}$  con respecto al voltaje de salida medido sobre el circuito.

Adicionalmente, al medir sobre el circuito se observó que el valor de la ganancia el cual debe ser 3.0 se encuentra en 2.89, esto se puede deber a que los componentes del circuito especialmente las resistencias que componen la ganancia no son lo suficientemente precisos para obtener el valor exacto de 3.0, otro factor puede ser el valor de la amplitud de las señales de PWM el cual debería estar en 3.3 V pero al medir se observó que llega hasta los 3.34 V.

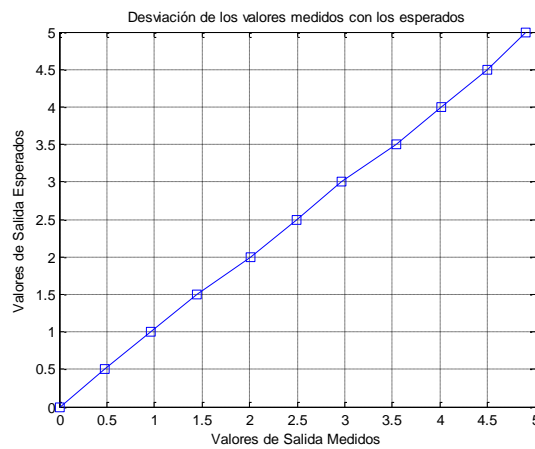


Figura 66. Desviación de los datos medidos con respecto a los valores esperados en la salida del circuito

En la Figura 66 se muestra la desviación que existe entre los valores medidos a la entrada de las UPM y los valores esperados, como se puede observar la relación tiene tendencia lineal y la mayoría de los datos están medidos están cercanos a los valores esperados, esto se pudo comprobar con el cálculo del error mostrado anteriormente.

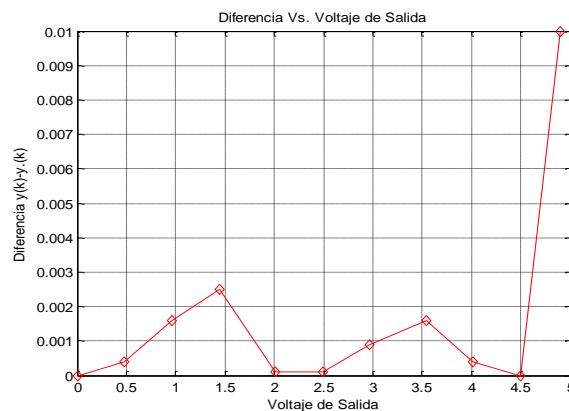


Figura 67. Gráfica al comparar los valores medidos y su diferencia con respecto a los valores esperados.

## 10.2 DISTANCIA ENTRE LOS MÓDULOS XBEE AL EJECUTAR EL CONTROL

A continuación se muestra una tabla con los resultados obtenidos al aumentar la distancia entre los módulos XBee ejecutando el controlador y permitiendo o no que existe línea de vista entre los módulos.

Tabla 16. Resultados de las pruebas de distancia.

Distancia (m)	Línea de Vista	Resultado
0,1	SI	EXITOSO
0,1	NO	EXITOSO
2,1	SI	EXITOSO
2,1	NO	NO EXITOSO
3,5	SI	EXITOSO
3,5	NO	NO EXITOSO

Al analizar estos resultados se nota que los módulos XBee pueden cumplir con su función de transmisión a distancias hasta 3,5 metros siempre y cuando la línea de vista entre los mismos no sea interrumpida, este es el caso de las distancias de 2,1 m y 3,5 m como se ve en la tabla la prueba no fue exitosa si no existía una línea de vista libre. Es de resaltar que para el caso de 0,1 m (10 cm) el hecho de que la línea de vista este interrumpida no afectó la trasmisión, esto se debe a que la distancia entre los módulos es demasiado corta y se hace irrelevante el bloqueo de la línea de vista.

## 10.3 RENDIMIENTO DEL SISTEMA DE CONTEO DE LOS ENCODERS

En las figuras que se muestran a continuación se podrá observar los resultados obtenidos en cuanto al rendimiento de los encoders variando la velocidad y el tamaño del buffer de envío del bloque stream server de QuaRC®.

En la figura 68 se ve la gráfica del rendimiento de los encoders versus el tamaño del buffer de envío a una velocidad para las velocidades de trasmisión 115.200, 57.00 y 38.400 Baudios. Para la velocidad de 115.200 Baudios puede deducir por el comportamiento de la gráfica que el rendimiento en general es muy bueno dando un rango desde el 95% hasta el 99%. Para tamaños de buffer menores a 70 el rendimiento del sistema se disminuye en gran medida, tanto que se pierden la mayoría de los datos y por ende no es posible utilizarlos ya que darían un resultado erróneo al implementar el control.

En la gráfica se muestran tamaños de buffer desde 70 hasta 600, para la gráfica azul en el intervalo entre 70 y 400 el sistema es capaz de contar los datos de forma efectiva y se tiene un promedio del 99.17% lo que quiere decir que a esta velocidad y para este rango de tamaño de buffer el sistema toma los datos casi en su totalidad, en el intervalo de 400 hasta 550 el rendimiento baja un poco y está en un promedio del 95.68% esto significa el sistema pierde algunos datos pero aun así se puede decir que el rendimiento sigue siendo muy bueno. De aquí en adelante el sistema tiende a mantenerse en un rendimiento del 98%.

En la figura, la gráfica de color rojo muestra el comportamiento del sistema de conteo para los encoders pero a una velocidad de trasmisión de 57.600 Baudios, en general el rendimiento es regular, como se observa el rendimiento del sistema para tamaños de buffer desde 70 hasta 130 es bueno y se encuentra en el 92% pero desde 130 hasta 280 el rendimiento baja rápidamente y se encuentra en un promedio del 32% lo cual indica que se están perdiendo demasiados datos y que si se va a utilizar esta velocidad de trasmisión no es recomendable utilizar tamaños de buffer en este rango.

A partir del tamaño de buffer de 310 se observa una mejoría considerable, el rendimiento pasa de un 16.47% a un 89.41% y se mantiene así hasta el tamaño de buffer 450 con un promedio del 90.6%, si se va a trabajar con esta velocidad de trasmisión es probable que la mejor opción sea utilizar tamaños de buffer

dentro de este rango, a pesar de que el rendimiento no sea tan bueno como en la velocidad anterior todavía se puede decir que el control lograría realizarse.

Para la gráfica de color verde se trabajó con una velocidad de transmisión de 38.400 baudios, como se observa el rendimiento del sistema del conteo de los encoders es el más bajo hasta ahora, su valor de rendimiento más alto es 82.35% y el más bajo es 21.17%.

En el primer rango de los tamaños de buffer de la gráfica (70 a 130) se tiene un rendimiento promedio del 76,8% el cual aunque es bajo no es del todo malo y sería todavía posible trabajar con el controlador, en el siguiente rango que comprende los tamaños de 160 a 350 se observa un rendimiento muy bajo del 34.12% como se había dicho anteriormente no es recomendable implementar el controlador con un rendimiento de este tipo ya que se han perdido demasiados datos y el controlador no funcionaría de forma eficiente o probablemente no se lograría estabilizar en ningún momento. En el último rango que comprende los tamaños de 400 a 600 el rendimiento del sistema logra subir un poco y mantenerse en un promedio del 48.23%, aunque es considerablemente mejor que el anterior intervalo la perdida de datos es todavía muy grande para poder implementar el controlador y que este funcione de forma correcta.

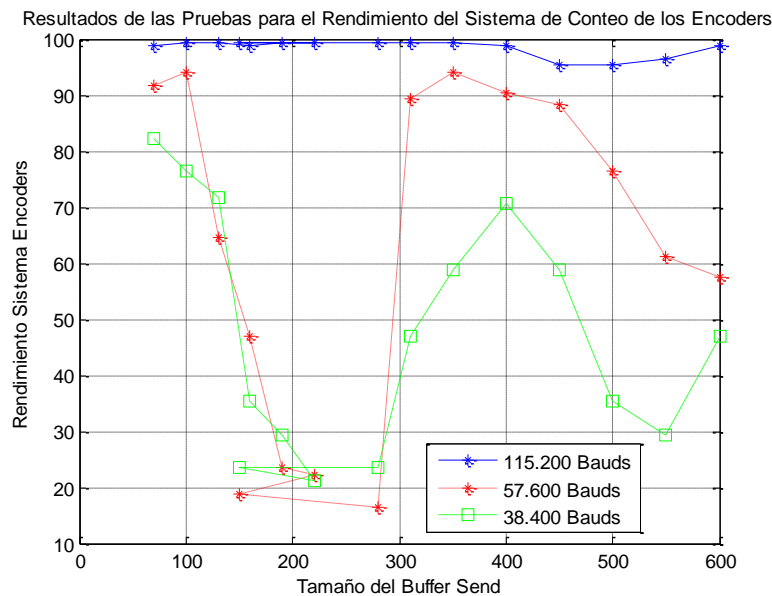


Figura 68. Rendimiento del sistema para conteo de los encoders para 3 diferentes velocidades

#### 10.4 ANALISIS DE PRUEBAS OBTENIDAS DE LAZO ABIERTO

De acuerdo con las gráficas obtenidas en la sección 9.4, se realizó la comparación de estos resultados con pruebas que se realizaron en la tarjeta Q8-USB de Quanser® y con las simulaciones del sistema ideal que se encuentran en el anexo 1.

En el caso en el que se aplica una señal de referencia de 2 V y 3 V en pitch, se tiene los siguientes resultados al comparar la tarjeta Q8-Usb y la tarjeta inalámbrica respectivamente:

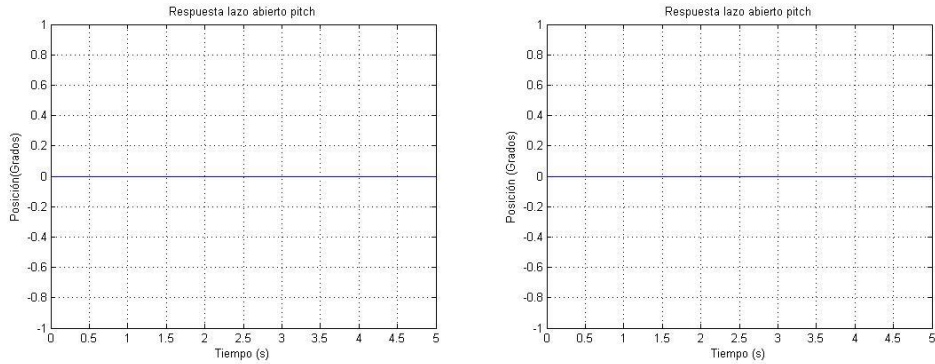


Figura 69. Respuesta de pitch lazo abierto con Q8-USB y tarjeta inalámbrica

Como era de esperarse en los dos casos, cuando se aplica un voltaje a la entrada de pitch sin variar el de yaw, la respuesta paso del sistema en pitch no tiende a variar su posición por lo que si no tiene un punto de referencia al realizar la simulación, el helicóptero no se moverá en este eje y por tanto no habrá efecto así la ganancia en pitch o yaw varíe.

Para el caso de la salida en yaw se obtuvieron los siguientes resultados con -2 y 3 de ganancia de pitch respectivamente:

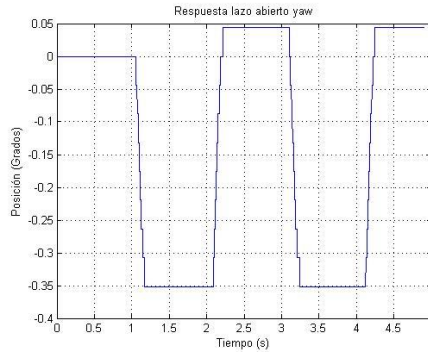


Figura 70. Respuesta de yaw lazo abierto con Q8-USB y tarjeta inalámbrica con ganancia de -2

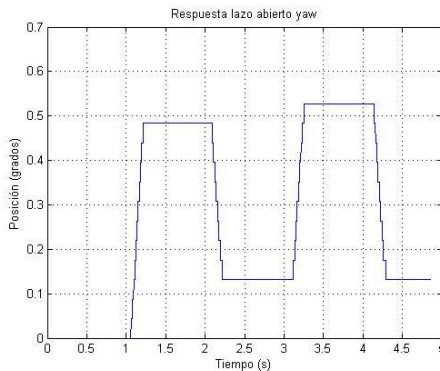


Figura 71. Respuesta de yaw lazo abierto con Q8-USB y tarjeta inalámbrica con ganancia de 3

En estos dos casos se puede observar que en las dos tarjetas, la posición del helicóptero en yaw varía respecto a la frecuencia de la señal de referencia en amplitudes muy pequeñas y además tiene un comportamiento creciente como se espera de la respuesta paso de yaw en lazo abierto ante un voltaje de pitch, donde es inestable ante esta excitación, como se puede observar en el anexo 1.

Por otro lado, se observa en la figura 51 que la respuesta de la tarjeta inalámbrica tiene picos de ruido que pueden afectar si se realiza un lazo de realimentación. Como se explicó anteriormente, estos picos se deben a pérdida de datos en la recepción de los conteos de los encoders que pueden causarse por ruido

debido al canal o al tamaño de la cola FIFO de la configuración del bloque Stream Server de QuaRC® en la recepción que no permite que en ese instante todos los datos sean actualizados debidamente, por lo que el algoritmo de concatenación y corrección de errores, no es lo suficientemente robusto para atenuar todos los picos que pueda tener esta señal.

## 10.5 ANALISIS DE PRUEBAS OBTENIDAS DE LAZO CERRADO

### 10.5.1 ANALISIS DE RESULTADOS EN LA COMUNICACIÓN

De acuerdo con los resultados obtenidos en la sección de pruebas, se pudo observar cómo se afectaba la estabilización del helicóptero no sólo al cambiar la velocidad de transmisión, sino al variar los tamaños de los buffer de transmisión y recepción en el bloque stream server de QuaRC®. Esto se debe a que el envío de las señales de control hacia el microcontrolador para ser convertidas en señales de PWM no se actualiza lo suficientemente rápido por ejemplo para tamaños grandes de buffer, puesto que este necesita llenar todo su espacio con una gran cantidad de datos antes de ser enviados, como se especifica en *máximum-Throughput*. Por tanto, la posición de los ejes del helicóptero no logra actualizarse en los tiempos requeridos para su estabilización.

Respecto a la velocidad de transmisión, cuando el módulo XBee está realizando una comunicación half-duplex a la tasa más alta que permite, la cual es 115200 baudios, muchos datos se pierden al tratar de enviar y recibir datos en el mismo instante, por lo que puede ocasionar colisiones no esperadas en el canal o el ruido correspondiente al canal inalámbrico afectan gravemente la información a esta tasa de transmisión y recepción, por lo que en esas condiciones no es recomendado por el fabricante trabajar a la tasa más alta.[29]

En las tablas correspondientes a la sección 9.5 se pueden encontrar los datos generados al variar los tamaños de buffer y las velocidades de transmisión

Una observación importante respecto a estas tablas, es que al implementar los cinco controladores, estos tienen un comportamiento similar a una tasa de 38400 baudios, diferenciados básicamente por la amplitud de las oscilaciones hasta el punto en el que a un tamaño de buffer determinado, el sistema perdía la estabilización requerida por los motores del helicóptero y ante tamaños muy pequeños, las señales de control que se envían a las UPM no se actualizan lo suficientemente rápido para enviar en el instante requerido el valor actual en el que se encuentra el control, por lo que no se logra estabilizar el helicóptero en la posición de las señales de referencia.

En el caso de 115200 baudios, el helicóptero nunca se llega a estabilizar, debido a que en esta velocidad, la respuesta en lazo cerrado tiende a ser oscilante en el eje pitch, por lo que rápidamente se pierde toda referencia y el helicóptero se desestabiliza en su totalidad.

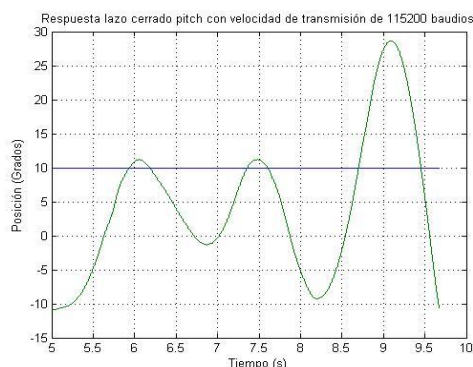


Figura 72. Respuesta de pitch en lazo cerrado con controlador LQR a una velocidad de 115200 baudios

## 10.5.2 COMPARACION CON MODELO TEORICO

Teniendo en cuenta el modelo teórico de la planta que se propuso en la sección 6.1 y que trae incorporado el sistema Demo de Quanser, se realizó una comparación del comportamiento dinámico y estático de las implementaciones realizadas en la sección de pruebas para el control del sistema con este modelo teórico, con el fin de poder observar las diferencias que se presentan en cada uno de los casos.

Para ello, se discretizó la planta del modelo teórico por medio de un retenedor de orden cero a la frecuencia de muestreo del sistema (1 kHz), con el fin de obtener una equivalencia entre este modelo y la implementación realizada por medio de la tarjeta de adquisición inalámbrica:

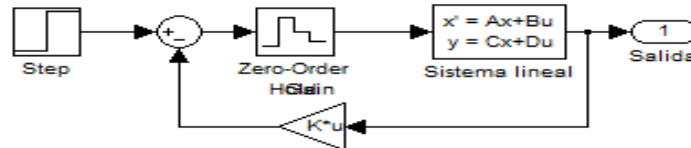


Figura 73. Diagrama en bloques del modelo teórico al ser discretizado con un zoh

La respuesta en lazo cerrado del modelo teórico ante una señal de referencia de 10 grados con un controlador LQR y un controlador LQI es la siguiente:

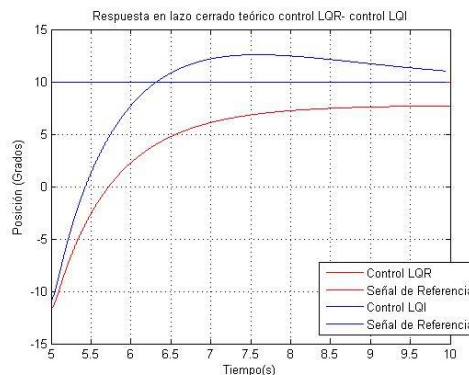


Figura 74. Respuesta en lazo cerrado teórico con control LQR y control LQI.

Las siguientes tablas muestran la comparación del comportamiento dinámico y estático del sistema, en el modelo teórico, la conexión entre la tarjeta Q8-USB de Quanser y la tarjeta inalámbrica y todo el sistema inalámbrico.

En la tabla 17 se observan las características del sistema cuando se realiza la conexión entre la tarjeta inalámbrica y la tarjeta Q8 USB, comparando el tiempo de establecimiento con el de la tabla 19 correspondiente al modelo teórico se puede observar que el valor promedio es 3.64 segundos mientras que en la tabla 17 el promedio es 3.72 segundos lo cual indica que la señal de control si se establece dentro de un rango tolerable y es posible que el helicóptero se estabilice. Otro parámetro importante es el error en estado estable, en la tabla 17 se tiene un promedio de 3.26 y en la tabla 19 se observa un promedio de 3.86, aquí se hace evidente que el error es menor en la conexión establecida para la tabla 17 pero aun así en sistema no es lo suficientemente preciso para que la salida en estado estable coincida con el valor de referencia esperado.

El tiempo de subida está en un promedio de 0.75 segundos para la tabla 17 y 2.25 segundos para la tabla 19, en general hasta ahora los valores obtenidos han sido similares a los valores del modelo teórico, el único factor que marca la diferencia y el cual afecta de forma muy significativa el rendimiento del sistema



se debe al máximo sobrepico, ya que como se observa en la tabla 19 se encuentra en un promedio de 2.4 % y en la tabla 17 se tiene un promedio de 31.8 % y es importante estas características sean lo menor posible para que el sistema pueda ser eficiente, es decir que si se lograra disminuir el valor del máximo sobrepico el sistema se estabilizaría mucho mejor.

Tabla 17. Características dinámicas y estáticas en lazo cerrado conexión Tarjeta Diseñada-Q8 USB

<b>Tarjeta Diseñada-Q8 USB</b>						
<b>Figura #</b>		<b>Máximo Sobrepico (Mp %)</b>	<b>Tiempo de Establecimiento (ts s)</b>	<b>Tiempo de Subida (tr s)</b>	<b>Tiempo de Pico Máximo (tp s)</b>	<b>Error de Estado Estable (Ess)</b>
<b>58</b>	kd1	37,5	3,5	0,53	1,1	7,5
	kd2	35,5	3,3	0,59	1,25	5
	kc	41,9	3	0,57	1,2	7,5
<b>54</b>	kd1	32,6	4,5	0,59	1,25	1,5
	kd2	31,2	3,5	1	1,3	5,2
	kc	26,6	4,3	0,58	1,15	1,5
<b>52</b>	LQI	45,4	3,3	1	1,4	3,5
	LQR	23,6	3	0,59	1,2	4,6
<b>56</b>	LQI	77,7	3,3	0,53	1,2	0,5
	LQR	30,2	3	0,58	1,15	7,5

Tabla 18. Características dinámicas y estáticas en lazo cerrado de la Tarjeta Diseñada

<b>Tarjeta Diseñada</b>						
<b>Figura #</b>		<b>Máximo Sobrepico (Mp %)</b>	<b>Tiempo de Establecimiento (ts s)</b>	<b>Tiempo de Subida (tr s)</b>	<b>Tiempo de Pico Máximo (tp s)</b>	<b>Error de Estado Estable (Ess)</b>
<b>63</b>	kd1	47,6	4	1,51	2,25	5
	kd2	31,8	3,5	1,52	2,25	7,5
	kc	68,1	4	1,5	2,15	10,3
<b>60</b>	LQI	59	4,2	1,58	2,5	4,5
	LQR	72,2	4,2	1,52	2,4	4,3

Para la tabla 18 se observa nuevamente el problema del máximo sobrepico, en este caso se tiene un valor del 55.7 % mientras que como ya se mencionó en la tabla 19 el valor corresponde al 2.4 %, las otras características se encuentran un poco más altas que los resultados de la tabla 17, pero aun así no están muy cercanos al modelo teórico y no presentan un inconveniente tan grande; la solución nuevamente radica en disminuir lo más posible todos los parámetros aquí descritos para que el sistema mejore su respuesta y se logre estabilizar de mejor forma.

Tabla 19. Características dinámicas y estáticas en lazo cerrado del Modelo Teórico

<b>Modelo Teórico</b>						
<b>Figura</b>		<b>Máximo Sobrepico (Mp %)</b>	<b>Tiempo de Establecimiento (ts s)</b>	<b>Tiempo de Subida (tr s)</b>	<b>Tiempo de Pico Máximo (tp s)</b>	<b>Error de Estado Estable (Ess)</b>
<b>74</b>	LQI	12,3	3,8	1,25	2,48	1,3
	LQR	0	3,6	2,5	0	4,5
<b>74</b>	kd1	0	3,6	2,5	0	4,5
	kd2	0	3,6	2,5	0	4,5
	kc	0	3,6	2,5	0	4,5

### 10.5.3 ANALISIS DEL EFECTO EN EL TIEMPO DE MUESTREO:

Al observar los resultados arrojados tanto por conexión Q8-Tarjeta inalámbrica como por todo el sistema inalámbrico, para los cinco controladores implementados al estabilizar el helicóptero, el comportamiento del sistema en lazo cerrado del sistema tiende a ser subamortiguado cuando logra estabilizarse, esto se debe principalmente a la tasa de muestreo a la cual está regida el sistema de control. Esto pudo ser comprobado al discretizar la planta con un retenedor de orden cero (zoh), al igual como se hizo en el modelo teórico, con el fin de observar el comportamiento del sistema ante variaciones del tiempo en el cual el retenedor de orden cero discretiza la planta, como se puede observar en el anexo 12.

Básicamente, variando la tasa a la cual funciona el retenedor de orden cero e implementando el controlador discreto a esa tasa de muestreo se pudieron obtener los siguientes resultados:

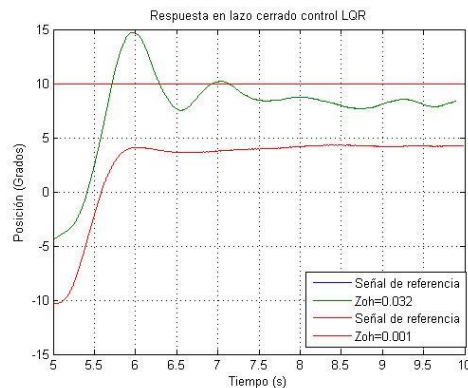


Figura 75. Respuesta de pitch en lazo cerrado con controlador LQR con Q8-USB con retenedor de orden cero con a) 0.001 s, b) 0.032 s

Como se puede observar en estas gráficas, al aumentar el tiempo en el cual el retenedor de orden cero toma cada muestra, la respuesta en lazo cerrado del helicóptero tiende a ser oscilante a tal punto que si se sigue aumentando esta tasa, el helicóptero se desestabilizará. Esto se debe a que a frecuencias de muestreo más bajas, el ancho de banda que requiere el helicóptero para su controlabilidad es muy cercano a este valor, por lo que a 0.032 s como tiempo de muestreo, el helicóptero podrá estabilizarse con las condiciones antes mostradas pero tendrá un comportamiento subamortiguado en su respuesta en lazo cerrado.

Por último, al observar las figuras correspondientes a la sección 9.6 del control con todo el sistema inalámbrico, es notable el ruido correspondiente a las señales de los encoders que se deben a picos con valores no coherentes a la respuesta del control. Como ya se dijo anteriormente, el algoritmo de codificación de datos corrige estos errores en su mayoría, pero existen otros muy difíciles de controlar como lo son ruido del canal y pérdida de datos en la comunicación, por lo que pueden llegar muchos datos erróneos de forma consecutiva y por tanto el sistema de codificación no podrá identificar si estos datos son ruido o no, entonces si hay una gran presencia de estos datos indeseados, el helicóptero puede llegar a desestabilizarse al no recibir el dato correcto en ese instante de tiempo.

## 11. ESTUDIO DE COSTOS

Este trabajo de grado fue sustentado con los recursos de los autores y de la carrera de Ingeniería Electrónica de la Pontificia Universidad Javeriana, a continuación se muestra una tabla donde se discriminan los costos reales y el valor aproximado a cobrar por el proyecto en caso de venderlo.

Durante la realización de este trabajo de grado se trabajó un promedio de 20 horas semanales durante 19 semanas lo que equivale a un total de 380 horas.

Tabla 20. Costos reales de los recursos utilizados para la realización del Trabajo de Grado.

Recurso	Unidad	Costo unidad	Total Unidades	Costo Total
Investigadores	Hora	20.000	380	7'600.000
Licenciamiento Software MATLAB® & SIMULINK	1 Licencia	21'950.000	1	21'950.000
Licenciamiento Software QuaRC®	1 Licencia	773.000	1	773.000
Componentes para el Circuito Impreso	Unidad	--	123	78.350 (Envío incluido)
Manufactura del Circuito Impreso	Unidad	31.250	4	145.000 (IVA incluido)
Uso de recursos del Laboratorio, Computador de la Facultad, Internet, etc.	Créditos-Semestre	375.400	6	2'252.400

## 12. TRABAJOS E INVESTIGACIONES A SEGUIR

El diseño e implementación de la tarjeta de adquisición tiene como finalidad ser utilizada y desarrollada en otros proyectos de investigación que abarquen la problemática de un control de forma inalámbrica. El primero de los temas a desarrollar en trabajos posteriores es la posibilidad de conectar en red varias tarjetas con algún protocolo de comunicación industrial como WirelessHart e ISA 100 que trabajan bajo el mismo estándar 802.15.4 y podrían manejar las mismas velocidades, de tal forma que desde cualquier computador que este en la red sea posible realizar el control del helicóptero asegurando un mejor control del flujo de datos.

Otro desarrollo a seguir muy importante es la implementación en MATLAB® de algún sistema de estimación y predicción de las señales provenientes de los encoders, como lo es un filtro de Kalman o un filtro de Wiener, que permitirán filtrar el ruido proveniente del canal inalámbrico. Además se podrá estudiar el efecto de retardos controlados en el sistema y las posibles variaciones que puede tener el sistema de control al evaluar distintas velocidades de transmisión, tiempos de muestreo, etc.

Dado que dos pines del microcontrolador han sido dispuestos para tener dos entradas análogas si se quiere probar el sistema con otra planta de Quanser®, se podría estudiar y evaluar leyes de control sobre estas, como lo son el motor DC o el péndulo invertido y poder observar el rendimiento que tiene la tarjeta sobre estos sistemas con los cambios pertinentes en la programación del microcontrolador y llegar a proponer mejoras a la tarjeta para un uso futuro en prácticas correspondientes a la sección de control.

### 13. CONCLUSIONES

La motivación de realizar un sistema de adquisición de datos inalámbrico para el helicóptero 2-DOF de Quanser®, surgió de la idea de integrar un protocolo de comunicación de este tipo y poder evaluar y estudiar el efecto que tenía sobre un sistema de control en específico, en este caso el helicóptero y así llegar a implementar en un ambiente de laboratorio este sistema sin restricciones de distancias o cableado respecto al computador donde se está ejecutando la acción de control.

El desarrollo de la tarjeta de adquisición permitió como primer gran logro la integración del protocolo de comunicación inalámbrica 802.15.4 del módulo XBee con la plataforma QuaRC® sin necesidad de conectar alguna tarjeta de Quanser®, a través de los bloques de comunicación que posee esta herramienta de Simulink, generando por tanto, una comunicación en tiempo real entre la planta y el controlador que se estaba ejecutando directamente desde Simulink.

Al diseñar la tarjeta de adquisición fue muy importante establecer los niveles de voltaje con los que trabaja el helicóptero y como se puede acoplar a los módulos de potencia UPM que posee Quanser® para enviar las señales de control correspondientes al helicóptero. Para lograr este diseño, primero se establecieron los parámetros del microcontrolador implementado, que es tal vez el elemento más importante de la tarjeta, puesto que mediante este, es posible realizar el conteo y envío de los pulsos provenientes de los encoders del helicóptero, que determinan la posición y dirección en la que se encuentra cada eje de la planta. Por otro lado, el microcontrolador también realiza la tarea de recibir los datos correspondientes a las señales de control de pitch y yaw del módulo XBee implementado en la tarjeta, demultiplexarlos en forma de señales PWM y enviarlas al circuito análogo, que de acuerdo a estas señales, fue diseñado para enviar las señales requeridas por el helicóptero.

El trabajo de grado dejó como resultado una estabilización del sistema inalámbrico pero con condiciones estrictas, como lo es velocidad de transmisión (38400 baudios) y parámetros de configuración en el bloque de comunicación con QuaRC®, teniendo como mayor restricción el sistema de conteo y adquisición de datos de las señales de los encoders, que para un futuro estudio, requiere mejoras. Además, el sistema requiere del diseño de un controlador que pueda minimizar las oscilaciones presentes en este estudio y en los controladores implementados y que tenga presente la tasa de muestreo final a la que se tomaron los datos provenientes de los encoders.

Integrando la tarjeta de adquisición de datos con QuaRC® y con el sistema de control para el helicóptero, se encontraron varias dificultades para lograr la estabilización del helicóptero con una ley de control ya existente, como lo fue acoplar la mejor tasa de transmisión de datos por el módulo XBee con los parámetros de envío y recepción que posee el bloque de QuaRC® con el que se realizó la implementación, para lograr con ello, un balance entre la eficiencia de los datos provenientes de los encoders y la estabilización del helicóptero. Esto conlleva a que bajo condiciones estrictas de los parámetros e implementando el controlador adecuado respecto al tiempo de muestreo con el que se rige el sistema, el helicóptero puede estabilizarse.

Otro factor muy importante a tener en cuenta en un sistema de control inalámbrico es la línea de vista de los módulos que se están comunicando. Para evitar pérdidas de datos en la transmisión inalámbrica es necesario asegurar que la línea de vista entre los módulos XBee siempre este libre, esto permitirá que la transmisión se realice de forma eficiente y garantizará la comunicación estable entre los módulos a pesar de la distancia siempre y cuando se encuentren dentro del radio de 10 metros establecidos en las especificaciones de la antena. Una solución es implementar un protocolo de comunicación inalámbrica más robusto, como WirelessHart, que se basa en el mismo estándar de Zigbee, permitiendo trabajar en las mismas velocidades de transmisión.

Los resultados obtenidos en este proyecto, pueden mejorarse al implementar un algoritmo de conteo de datos más robusto en el microcontrolador, de tal forma que optimice la tasa de conteo a la velocidad de transmisión sin perder información de los pulsos de los encoders en cuadratura. Por tanto, se pueden obtener mejores resultados a velocidades de transmisión bajas y puede ser probado en condiciones más amplias y menos estrictas ante una ley de control logrando un rendimiento similar al que se obtiene con una tarjeta de Quanser®.

Como ya se dijo anteriormente, este estudio tiene como finalidad servir de base para futuros proyectos ligados al área de control y que llegue a optimizarse de tal forma que pueda ser una gran herramienta para prácticas de laboratorio relacionadas a las asignaturas de control de la Pontificia Universidad Javeriana, no solamente para el estudio de controladores sobre el helicóptero 2-DOF, sino también para las otras plantas de Quanser® que posee la facultad de Ingeniería Electrónica.

#### 14. BIBLIOGRAFIA Y FUENTES DE INFORMACION

- [1] J. H. Taylor and H. M. S. Ibrahim, “*A new, Practical Approach to maintaining an Efficient Yet Acceptably-performing Wireless Networked Control System*”. In: 2012 International Conference on System Science and Engineering.
- [2] Mandos de Vuelo. [En línea]. Disponible en: [http://es.wikipedia.org/wiki/Mandos\\_de\\_vuelo](http://es.wikipedia.org/wiki/Mandos_de_vuelo)
- [3] R. A. Gupta and M. Y. Chow, “*Networked Control System: Overview and Research Trends*”. In: IEEE Transactions On Industrial Electronics, Vol. 57, No. 7, July 2010.
- [4] M. S. Hasan, H. Yu and A. Carrington, “*Overview of Wireless Networked Control Systems over Mobile Ad-hoc Network*”. In: Proceedings of the 14th International Conference on Automation & Computing, Brunel University, West London, UK, 6 September 2008.
- [5] Y. A. Millán, F. Vargas, F. Molano and E. Mojica, “*A Wireless Networked Control Systems Review*”. In: IEEE, 2011.
- [6] J. E. Nagel and C. F. Walker, “*Design and Implementation of a Wireless Data Acquisition System*”. In: IEEE, 2003.
- [7] M. Parada, J. I. Escudero and P. Simón: “*Apuntes de Instrumentación, Técnicas de Medida y Mantenimiento*”. In: Facultad de Informática y Estadística, Sevilla.1998.
- [8] D. Libo, Z. He, L. Haojie and D. Hongji, “*Design and Implementation of a Small-Size Wireless Data Acquisition System*”. In: IEEE Instrumentation and Measurement, Technology Conference. Anchorage, AK, USA, 21-23 May, 2002.
- [9] J. M. Moreno and D. R. Fernández, “*Informe Técnico: Protocolo ZigBee (IEEE 802.15.4)*”. In: June, 2007.
- [10] Quanser, “*2-DOF Helicopter, Reference Manual*”. In: Quanser Reference Manual, No. 2.2, 4-22.
- [11] Microchip, “*PIC18F2331/2431/4331/4431 Data Sheet*”. In: Microchip Technology, USA, 4-16, 147-148, 2003.
- [12] Quanser Consulting, “*Q4 Data Acquisition System User’s Guide Version 1.0*”. In: Canada, 20, 2003.

- [13] J. Venegas. “*Encoders*”. In: Departamento de Electrónica, Universidad Técnica Federico Santa María, Valparaíso, Chile, 1-3, May, 2009.
- [14] MaxStream. “XBee™/XBee-PRO™ OEM RF Modules Product Manual v1.xAx – 802.15.4 Protocol”. In: MaxStream Inc., 2006.
- [15] FTDI Chip “FT232R USB UART I.C.”. In: Future Technology Devices International Ltd., 2005.
- [16] A. Palacherla, “*Using PWM to Generate Analog Output*”. In: Microchip Technology Inc. USA, 1-2, 1997.
- [17] Texas Instruments, “*LMV321/LMV358/LMV324 Single/Dual/Quad General Purpose, Low Voltage, Rail-to-Rail Output Operational Amplifiers*”. In: Texas Instruments Inc. USA, 1-7, 2004.
- [18] Intersil, “*60MHz Rail-to-Rail Input-Output Operational Amplifier EL5211T*”. In: Intersil Americas Inc. USA, 1-5, May, 2010.
- [19] Quanser Consulting, “*Universal Power Module 1503, 1506, 2405, 2405-PWM, User Manual*”. In: Canada, 1-3.
- [20] Digi. “X-CTU Configuration & Test Utility Software User’s Guide”. In: Digi International, 2008.
- [21] Digi International. [En línea]. [Consultado 17 de Marzo de 2013]. Disponible en Internet: <http://www.digi.com/support/kbase/kbaseresultdetl?id=2213>.
- [22] QuaRC Service. [En línea]. [Consultado 24 de Marzo de 2013]. Disponible en Internet: [http://www.QuaRCservice.com/ReleaseNotes/files/release\\_notes\\_index.html](http://www.QuaRCservice.com/ReleaseNotes/files/release_notes_index.html).
- [23] QuaRC Service. [En línea]. [Consultado 24 de Marzo de 2013]. Disponible en Internet: [http://www.QuaRCservice.com/ReleaseNotes/files/serial\\_protocol.html](http://www.QuaRCservice.com/ReleaseNotes/files/serial_protocol.html).
- [24] Lcardaba. [En línea]. [Consultado 25 de Marzo de 2013]. Disponible en Internet: <http://www.lcardaba.com/articles/opamps.html>.
- [25] Unicrom. [En línea]. [Consultado 19 de Marzo de 2013]. Disponible en Internet: [http://www.unicrom.com/Tut\\_OpAmpNoInversor.asp](http://www.unicrom.com/Tut_OpAmpNoInversor.asp).
- [26] Quanser, “*QuaRC Controls Development Made Easy*”. In: Quanser Product Information Sheet, No. B, pp. 1-2.
- [27] Wordpress. [En línea]. [Consultado 24 de Marzo de 2013]. Disponible en Internet: <http://galaxi0.wordpress.com/el-puerto-serial/>
- [28] A. Cruz, E. Portilla, R. Silva. “*Multiplicador Electrónico para Encoder Incremental*”. In: Cenrto de innovación y Desarrollo Tecnológico del Instituto Politécnico Nacional, México D.F, México, 4, November, 2008.
- [29] Digi International. [En línea]. [Consultado 17 de Marzo de 2013]. Disponible en Internet: <http://www.digi.com/support/forum/4787/using-the-xbee-at-115-200-baud-updated-16-march-2010>