

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

**UNIDAD ZACATENCO**

**DEPARTAMENTO DE CONTROL AUTOMÁTICO**

**“Síntesis de un esquema de linealización  
robusta por realimentación de estado  
estructural para un cuadricóptero”**

**T E S I S**

Que presenta

**Luis Ángel Blas Sánchez**

Para obtener el grado de

**DOCTOR EN CIENCIAS**

EN LA ESPECIALIDAD DE

**Control Automático**

Directores de la Tesis:

**Dr. Moisés Bonilla Estrada**

**Dr. Sergio Rosario Salazar Cruz**

Ciudad de México

Enero, 2020



## **Agradecimientos**

Inicialmente quiero agradecer a mi familia por todo su apoyo.

A todos mis amigos y compañeros por esos momentos compartidos.

Agradezco a mis asesores de tesis, el Dr. Moisés Bonilla y el Dr. Sergio Salazar por su apoyo en la realización de esta tesis.

Agradezco al Consejo Nacional de Ciencia y Tecnología por el apoyo económico recibido durante todo el programa de estudio.

Finalmente agradezco al Cinvestav por todo su apoyo.



## Resumen

En este trabajo de tesis se considera una clase de sistemas de control afines y se propone una técnica de linealización por realimentación estructural. Este enfoque relativamente simple tiene la ventaja de combinar las técnicas de control lineal clásicas con las técnicas de control robusto.

El esquema de control propuesto es especialmente adecuado para vehículos aéreos no tripulados, donde es importante rechazar no solo las no linealidades y los cambios inesperados de la estructura, sino también buscar la simplicidad y efectividad del esquema de control para su programación en el piloto automático a bordo de la aeronave.



## **Abstract**

In this thesis work, we consider a class of affine control systems and we propose a structural feedback linearization technique. This relatively simple approach has the advantage of combining the classical linear control technique with the robust control technique. The proposed control scheme is especially suitable for unmanned aerial vehicles, where it is important not only to reject the nonlinearities and the unexpected changes of the structure, but also to look for the simplicity and effectiveness of the control scheme for programming in the autopilot on board the aircraft.





# Índice general

Agradecimientos	iii
Resumen	v
Abstract	vii
Índice general	ix
Índice de figuras	xv
<b>1 Introducción</b>	<b>3</b>
1.1 Estado del arte . . . . .	4
1.2 Objetivo general . . . . .	6
1.2.1 Objetivos particulares . . . . .	6
1.3 Organización del trabajo . . . . .	7
<b>2 Modelado del cuadricóptero</b>	<b>9</b>
2.1 Descripción del cuadricóptero . . . . .	9
2.2 Formulación de Euler-Lagrange . . . . .	11
2.3 Modelo incremental . . . . .	19

2.4	Representaciones de estado . . . . .	21
2.5	Conclusiones . . . . .	23
<b>3</b>	<b>Linealización</b>	<b>25</b>
3.1	Linealización aproximada . . . . .	25
3.2	Linealización exacta entrada-salida . . . . .	28
3.2.1	Descomposición estructural . . . . .	29
3.2.2	El esquema exacto de la linealización por realimentación estructural . . . . .	31
3.3	Linealización exacta entrada-salida del cuadricóptero . . . . .	33
3.3.1	Realimentaciones de estado . . . . .	33
3.3.2	Descomposición estructural . . . . .	36
3.4	Conclusiones . . . . .	39
<b>4</b>	<b>Linealización robusta asintótica por realimentación</b>	<b>41</b>
4.1	Estimación de la no linealidad . . . . .	41
4.1.1	Sistema en lazo cerrado . . . . .	43
4.2	Simulación numérica . . . . .	45
4.2.1	Realimentaciones de estado . . . . .	46
4.2.2	Filtro Beard-Jones . . . . .	47
4.2.3	Filtro pasa altas del sistema en lazo cerrado . . . . .	54
4.2.4	Resultados de simulación . . . . .	56
4.2.5	Respuesta en frecuencia . . . . .	61
4.2.6	Alargamiento de la vecindad de estabilidad . . . . .	61
4.3	Conclusiones . . . . .	67

---

<b>5 Experimentos en exteriores: Vuelo estacionario</b>	<b>69</b>
5.1 Realimentación localmente estabilizante . . . . .	69
5.2 Estimador de señal de incertidumbre no lineal . . . . .	71
5.2.1 Relación entre los polinomios característicos . . . . .	72
5.2.2 Respuesta en frecuencia . . . . .	73
5.2.3 Estimación del ancho de banda . . . . .	74
5.3 Vuelo estacionario en exteriores . . . . .	75
5.3.1 Subsistema $x - \theta$ . . . . .	76
5.3.2 Subsistema $y - \phi$ . . . . .	78
5.3.3 Subsistema $\psi$ . . . . .	80
5.3.4 Subsistema $z$ . . . . .	81
5.4 Conclusiones . . . . .	82
<b>6 Seguimiento de trayectoria</b>	<b>85</b>
6.1 Ruido del viento . . . . .	85
6.2 Fenómeno de deriva . . . . .	88
6.3 Estimación sin deriva . . . . .	89
6.3.1 Estimador sin deriva para el subsistema $x - \theta$ : Forma observador . . . . .	90
6.3.2 Estimador sin deriva para el subsistema $y - \phi$ : Forma observador . . . . .	90
6.3.3 Estimador sin deriva para el subsistema $\psi$ : Forma observador . . . . .	90
6.4 Sistema en lazo cerrado . . . . .	91
6.4.1 Función de transferencia . . . . .	91
6.4.2 Respuesta en frecuencia . . . . .	92
6.5 Seguimiento de trayectoria . . . . .	93

6.5.1	Esquema de control . . . . .	94
6.5.2	Planificación de trayectoria de estado óptima . . . . .	94
6.5.3	Partición de la trayectoria circular . . . . .	96
6.6	Experimentos en exteriores . . . . .	100
6.6.1	Trayectoria lenta . . . . .	101
6.6.2	Trayectoria rápida . . . . .	105
6.7	Conclusiones . . . . .	111
<b>7</b>	<b>Conclusiones</b>	<b>113</b>
7.1	Lista de publicaciones . . . . .	116
<b>A</b>	<b>Cálculo de las constantes <math>k_F</math> y <math>k_T</math></b>	<b>117</b>
<b>B</b>	<b>Matriz de inercia del cuadricóptero</b>	<b>119</b>
<b>C</b>	<b>Componentes del cuadricóptero</b>	<b>123</b>
C.1	Estructura mecánica del cuadricóptero . . . . .	123
C.2	Sistema de propulsión . . . . .	125
C.2.1	Hélices . . . . .	125
C.2.2	Motores brushless . . . . .	127
C.2.3	Controlador Electrónico de Velocidad (ESC) . . . . .	128
C.2.4	Batería . . . . .	130
C.3	Sistema de control y comando . . . . .	131
C.3.1	Transmisor DX7s . . . . .	131
C.3.2	Piloto automático Pixhawk . . . . .	133
C.4	El módulo UBlox GPS + Compass . . . . .	136

C.5	Otros componentes . . . . .	137
C.5.1	Arnés de distribución de energía . . . . .	137
C.5.2	Módulo de alimentación . . . . .	138
C.5.3	Cargador de baterías Lipo . . . . .	139
<b>D</b>	<b>Código del cuadricóptero</b>	<b>141</b>
D.1	Instalación de las herramientas de compilación . . . . .	141
D.2	Mission Planner . . . . .	147
D.3	Cargar el firmware personalizado al Pixhawk . . . . .	149
D.4	Descripción del código personalizado . . . . .	154
D.4.1	Programa principal . . . . .	156
D.4.1.1	Archivos de cabecera (Headers) . . . . .	156
D.4.1.2	Definición de variables y funciones nativas . . . . .	158
D.4.1.3	Planificador de tareas . . . . .	162
D.4.1.4	void setup() . . . . .	163
D.4.1.5	void loop() . . . . .	163
D.4.1.6	Funciones auxiliares . . . . .	164
D.4.1.7	AP_HAL_MAIN(); . . . . .	164
D.4.2	Funciones del programa principal . . . . .	165
D.4.2.1	Variables globales . . . . .	165
D.4.2.2	Parámetros para el Pixhawk . . . . .	170
D.4.2.3	Lectura del radiocontrol . . . . .	172
D.4.2.4	Lectura del GPS . . . . .	173
D.4.2.5	Lectura del barómetro . . . . .	175

D.4.2.6	Guardar datos en la memoria micro SD . . . . .	175
D.4.2.7	Iniciar ArduPilot . . . . .	176
D.4.2.8	Fast loop . . . . .	177
D.4.3	Funciones de funciones del programa principal . . . . .	179
D.4.3.1	Generar origen arbitrario . . . . .	179
D.4.3.2	Paquetes de datos . . . . .	181
D.4.3.3	Realimentaciones de estado . . . . .	187
D.4.3.4	Filtros Beard-Jones . . . . .	189
D.4.3.5	Generación de trayectorias . . . . .	195
<b>E</b>	<b>Publicaciones</b>	<b>203</b>
	<b>Bibliografía</b>	<b>253</b>

# Índice de figuras

2.1	Diagrama de conexión de la plataforma experimental. . . . .	10
2.2	Marcos de referencia. . . . .	11
2.3	Diagrama para obtener los pares. . . . .	15
4.1	Diagrama a bloques de la función de transferencia en lazo cerrado. . . . .	45
4.2	Diagrama del lugar geométrico de las raíces. . . . .	49
4.3	Gráfica de Bode de magnitud. . . . .	55
4.4	Respuestas obtenidas para el subsistema $x - \theta$ . . . . .	57
4.5	Respuestas obtenidas para el subsistema $y - \phi$ . . . . .	58
4.5	Respuestas obtenidas para el subsistema $y - \phi$ (continuación). . . . .	59
4.6	Respuestas obtenidas para el subsistema $\psi$ . . . . .	59
4.6	Respuestas obtenidas para el subsistema $\psi$ (continuación). . . . .	60
4.7	Respuestas obtenidas para el subsistema $z$ . . . . .	60
4.8	Rechazo de la señal de incertidumbre no lineal. . . . .	61
4.9	Comportamiento de la constante de Lipschitz. . . . .	62
4.10	Acotamiento de $\tilde{q}_*$ bajo razón de cambio acotada. . . . .	63
4.11	Propiedades de los difeomorfismos. . . . .	64
4.12	Estabilidad. . . . .	64

4.13 Acotamiento de $\tilde{q}_*$ bajo ancho de banda limitada. . . . .	65
4.14 Atracción de las trayectorias $x(t)$ a la vecindad. . . . .	65
4.15 Acotamiento de $\tilde{q}_*$ . . . . .	66
4.16 Alargamiento de la vecindad de estabilidad . . . . .	67
5.1 Gráfica de Bode. . . . .	74
5.2 Posición en $y$ y espectro de amplitud. . . . .	75
5.3 Gráficas de vuelo estacionario del subsistema $x - \theta$ . . . . .	77
5.4 Comparación de las señales de posición $x$ . . . . .	78
5.5 Gráficas de vuelo estacionario del subsistema $y - \phi$ . . . . .	79
5.6 Comparación de las señales de posición $y$ . . . . .	80
5.7 Gráficas del subsistema $\psi$ . . . . .	81
5.8 Gráficas del subsistema $z$ . . . . .	82
6.1 Caracterización del ruido del viento. . . . .	87
6.2 Fenómeno de deriva. . . . .	88
6.3 Gráfica de Bode. . . . .	93
6.4 Simulación numérica para una partición gruesa para el subsistema $x - \theta$ . . .	97
6.5 Simulación numérica para una partición gruesa para el subsistema $y - \phi$ . . .	98
6.6 Seguimiento de una trayectoria circular en el plano $(x, y)$ con 5 m de radio en 60 s ( $N = 60$ y $T_s = 1$ s). . . . .	103
6.7 Comparación del error de seguimiento. . . . .	104
6.8 Seguimiento de una trayectoria circular en el plano $(x, y)$ con 5 m de radio en 20 s ( $N = 20$ y $T_s = 1$ s). . . . .	106
6.9 Comparación del error de seguimiento. . . . .	107



6.10	Altura $z$ y ángulo de guiñada $\psi$ en seguimiento lento. . . . .	108
6.11	Altura $z$ y ángulo de guiñada $\psi$ en seguimiento rápido. . . . .	109
6.12	Acciones de control $u_x$ y $u_y$ para el seguimiento rápido de trayectoria. . . . .	110
6.13	Acciones de control $u_z$ y $u_\psi$ para el seguimiento rápido de trayectoria. . . . .	111
B.1	Prisma. . . . .	120
B.2	Cilindro. . . . .	120
B.3	Plataforma experimental utilizada en esta tesis. . . . .	121
C.1	Piezas de la estructura del cuadricóptero. . . . .	124
C.2	Estructura del cuadricóptero. . . . .	124
C.3	Hélices. . . . .	126
C.4	Características de las hélices. . . . .	126
C.5	Motor 4220-880Kv . . . . .	127
C.6	Controlador de velocidad. . . . .	129
C.7	Batería Lipo. . . . .	130
C.8	Transmisor DX7s. . . . .	132
C.9	Pixhawk. . . . .	133
C.10	Piloto automático Pixhawk y componentes. . . . .	135
C.11	GPS con módulo de brújula. . . . .	136
C.12	Arnés de distribución de energía. . . . .	138
C.13	Módulo de alimentación. . . . .	139
C.14	Cargador de baterías IMAX. . . . .	140
D.1	Consola PX4. . . . .	142
D.2	Firmware creado en el directorio ArduCopter. . . . .	143

D.3	Ventana de Eclipse. . . . .	144
D.4	Usar espacios en lugar de pestañas. . . . .	145
D.5	“K&R Tab”. . . . .	146
D.6	Seleccionando el tipo de archivo. . . . .	147
D.7	Interfaz de inicio de la estación de control de tierra Mission Planner. . . . .	149
D.8	Inicio de la instalación del firmware personalizado. . . . .	150
D.9	Conexión USB del Pixhawk. . . . .	150
D.10	Instalación del firmware personalizado. . . . .	151
D.11	Instalar la solicitud de firmware. . . . .	152
D.12	Progreso de carga del firmware. . . . .	153
D.13	Firmware cargado en el Pixhawk. . . . .	154
D.14	Esquema general del programa personalizado. . . . .	155



# Capítulo 1

## Introducción

Los avances en tecnología de microcomputadoras y la miniaturización de sensores han hecho realidad la construcción de vehículos aéreos no tripulados (UAV<sup>1</sup>). Un vehículo aéreo no tripulado (también conocido como dron) es una aeronave pequeña diseñada para volar sin un piloto a bordo. En el contexto de este trabajo, un UAV es una aeronave propulsada que un ser humano puede operar a distancia de forma manual o semiautónoma por control remoto, de forma autónoma mediante el uso de una computadora a bordo que puede pre-programarse para realizar una tarea específica, conjunto de tareas y también se puede operar mediante una estación de control en tierra que envía comandos de forma inalámbrica a la aeronave.

Un tipo de vehículos aéreos de ala rotatoria que en los últimos años han recibido considerable atención son los cuadricópteros. Estos pequeños helicópteros no tripulados tienen la capacidad de despegue y aterrizaje vertical (VTOL<sup>2</sup>) y vuelo estacionario. Debido a su tamaño compacto, alta maniobrabilidad y vuelo autónomo, se han convertido en una plataforma estándar para la investigación de robótica aérea en todo el mundo.

Los vehículos aéreos no tripulados se pueden utilizar en muchas aplicaciones como son: ciencias de la tierra [1], búsqueda y rescate [2], vigilancia de fronteras [3], inspección industrial [4], agricultura [5], investigación, etc. Esto se debe a su facilidad de despliegue, bajo costo

---

<sup>1</sup>Por sus siglas en inglés unmanned aerial vehicle.

<sup>2</sup>Por sus siglas en inglés Vertical Take-Off and Landing.

de mantenimiento y capacidad de vuelo estacionario.

Los vehículos aéreos no tripulados se han introducido a nivel académico como el tema de proyectos de investigación. El modelo dinámico básico del cuadricóptero es el punto de partida para todos los estudios realizados. También se han obtenido modelos dinámicos más completos al incluir la dinámica del motor y efectos aerodinámicos [6, 7].

### 1.1 Estado del arte

Recientemente, ha habido un gran interés en encontrar esquemas de control simples y efectivos para vehículos aéreos no tripulados (UAV), capaces de rechazar las no linealidades y los cambios de estructura inesperados. Estos esquemas de control deben ser simples y efectivos, ya que deben programarse en el piloto automático incorporado al cuadricóptero.

En la literatura se puede consultar las diferentes técnicas de control que se han evaluado, entre ellas están los controladores PD [8, 9], PID [10, 11], control “backstepping” [12, 13], control no lineal  $\mathcal{H}_\infty$  [14], controlador LQR [10], control por modos deslizantes y controladores no lineales con saturaciones anidadas [15, 16].

Un esquema de control natural y simple para estos sistemas no lineales es utilizar la aproximación clásica de Taylor. Este es de hecho un enfoque muy simple. Sin embargo, las técnicas correspondientes deben restringirse a una pequeña vecindad de un punto de referencia fijo. Bajo algunas suposiciones restrictivas relacionados con la perturbación externa, esta técnica puede ser útil para clases de problemas de regulación. Por otro lado, no se recomienda, por ejemplo, en problemas de control en seguimiento.

Otro esquema de control simple es la técnica de linealización exacta entrada-salida. Hay que tener en cuenta que este enfoque analítico requiere un conocimiento completo de los parámetros del modelo dinámico, así como de las derivadas correspondientes.

En [17], los autores presentan dos tipos de controladores no lineales para un cuadricóptero autónomo. Sin embargo, programar este esquema de control en el piloto automático incorpo-

rado al cuadricóptero es bastante difícil debido a las estimaciones de las derivadas necesarias.

En [18], los autores presentan diseños de control de seguimiento de orientación y trayectoria basados en una estructura de control de lazo interno/externo para condiciones de vuelo normales. Una vez más, programar este esquema de control en el piloto automático incorporado al cuadricóptero es bastante difícil debido a las estimaciones de las derivadas.

En [19], los autores proponen un controlador robusto basado en el enfoque de separación de escala de tiempo para lograr el despegue automático, vuelo estacionario, seguimiento de trayectoria y las misiones de aterrizaje para un cuadricóptero. Los autores solo muestran resultados experimentales en ambientes interiores.

Un esquema de control eficiente es aquel en el que las no linealidades se tratan como perturbaciones y luego se rechazan. En [20] se presenta una solución de realimentación de estado al problema de estabilizar un cuadricóptero a lo largo de una trayectoria predefinida en presencia de perturbaciones de fuerza constante. Estas perturbaciones se estiman a través del uso de “backstepping adaptable”. Los autores presentan resultados experimentales realizados en interiores y utilizando un sistema de captura de movimiento; agregan perturbaciones de viento de un sistema de aire acondicionado. Los autores afirman que su controlador propuesto es implementable solo en situaciones donde las mediciones del estado completo están disponibles para realimentación, y esta no es una circunstancia típica en la práctica para los cuadricópteros que se encuentran fuera de entornos controlados.

En [21], los autores presentan un esquema de control de rechazo activo de perturbación de doble lazo cerrado para tratar algunos problemas de control difíciles en el cuadricóptero como la no linealidad, el acoplamiento fuerte y la sensibilidad a la perturbación. Su esquema de control, al principio, desacopla el sistema de vuelo del cuadricóptero por medio de algunas variables de control virtual. Luego, aplican un algoritmo diferenciador de seguimiento no lineal cuyo objetivo es obtener una señal de entrada suave para reducir el error inicial del sistema. Y finalmente, estiman las perturbaciones internas desconocidas y externas, para compensarlas con una ley de control de realimentación de error de estado. Los autores presentan resultados de simulación para demostrar que su esquema de control puede garantizar que el

cuadricóptero sigue cierta trayectoria objetivo de forma rápida y precisa mientras mantiene la estabilidad, incluso con perturbaciones externas.

En [22], los autores presentan un controlador de orientación del cuadricóptero basado en el observador de perturbaciones para maniobras agresivas. El controlador está formado por la conexión en cascada entre dos lazos de control: un lazo de control de orientación externo basado en cuaterniones y un lazo de control de seguimiento de velocidad angular interno basado en el observador de perturbaciones. El controlador externo es una realimentación proporcional de la orientación relativa entre el marco fijo del cuerpo y el marco de referencia de orientación deseado, y el observador de perturbaciones interno está diseñado para estimar y compensar el término de Coriolis y las perturbaciones externas. Llevaron a cabo pruebas de vuelo para validar los resultados principales, mostrando un rendimiento de seguimiento sobresaliente cuando se ejecutan maniobras de orientación agresiva, incluso en presencia de perturbaciones fuertes como cargas útiles suspendidas.

## 1.2 Objetivo general

Desarrollar e implementar un esquema de control lineal capaz de rechazar las no linealidades del modelo y compensar los cambios inesperados de la estructura. Además debe ser simple, efectivo y fácil de programar en el piloto automático a bordo del cuadricóptero.

### 1.2.1 Objetivos particulares

- Construcción de un cuadricóptero para poner en funcionamiento el esquema de control.
- Sintetizar el esquema de control lineal.
- Realizar pruebas de vuelo estacionario en exteriores que involucren el esquema de control propuesto.
- Realizar pruebas de seguimiento de trayectoria en exteriores que involucren el esquema

de control propuesto.

- Comparar los resultados experimentales con un controlador por realimentación de estado.

## 1.3 Organización del trabajo

El resto de este trabajo está dividido de la siguiente manera.

En el capítulo 2 se deriva el modelo matemático del cuadricóptero mediante la formulación de Euler-Lagrange. Posteriormente se obtiene el modelo incremental y finalmente una representación en espacio de estados.

En el capítulo 3 se obtiene un modelo lineal aproximado del cuadricóptero bajo la consideración de ángulos pequeños. Posteriormente abordamos el método de linealización que se basa en la estructura interna de la descripción de estado del sistema linealizado, se presentan los conceptos necesarios para realizar este enfoque de linealización, se introduce un cambio de variable que mapea las no linealidades del sistema en una señal de incertidumbre no lineal  $q_*$ , contenida en el dominio de la acción de control  $u$ . El capítulo finaliza con la aplicación de este método de linealización al cuadricóptero.

En el capítulo 4 se muestra la forma de estimar la señal de incertidumbre no lineal,  $q_*$ . Para esto se utiliza el filtro Beard-Jones, el cual se basa en observadores de estado estándar. En la simulación numérica se aborda el cálculo de las matrices de ganancia de las realimentaciones de estado mediante el diseño LQR, el cálculo de las inyecciones de salida de los filtros Beard-Jones, la representación en la forma observador de los filtros Beard-Jones y los resultados de simulación. Finalmente se describe el alargamiento de la vecindad de estabilidad.

En el capítulo 5 se presenta un procedimiento de sintonización de los filtros Beard-Jones y se muestran resultados experimentales de vuelo estacionario en exteriores, aplicando el estimador de incertidumbre no lineal.



En el capítulo 6 se aborda el seguimiento de trayectoria en exteriores. Primero se analiza y caracteriza el fenómeno de deriva. Se obtiene el estimador de incertidumbre no lineal sin deriva. Se describe la síntesis de la trayectoria de estado óptima y se presentan resultados experimentales de seguimiento.

En el capítulo 7 se concluye.

# Capítulo 2

## Modelado del cuadricóptero

En este capítulo se desarrollará el modelado del cuadricóptero mediante la formulación matemática de Euler-Lagrange, la cual está basada en el concepto de energía mecánica (cinética y potencial). A partir de las ecuaciones que describen al sistema cuadricóptero, se obtendrá el modelo incremental. El capítulo se termina con la descomposición del modelo incremental en cuatro subsistemas y sus correspondientes representaciones de estado.

### 2.1 Descripción del cuadricóptero

El esquema general de conexión de la plataforma experimental se muestra en la figura 2.1. En esta figura se muestra que el cuadricóptero está constituido de la siguiente manera

- El cuadricóptero utilizado en los experimentos está construido en una estructura de fibra de carbono con una longitud de 498 mm.
- El controlador de vuelo Pixhawk<sup>TM</sup> tiene las siguientes características: un microprocesador de 32 bits STM32F427 Cortex-M4F con FPU, 168 MHz, 256 KB de RAM y 2 MB Flash. Cuenta con las interfaces I2C, PWM, puertos seriales 5x UART, dos entradas ADC, entradas Spektrum DSM y Futaba. Tiene integrado los siguientes sensores: magnetómetro, barómetro, dos acelerómetros y dos giroscopios.

- Usa un GPS externo Ublox Neo-M8N con brújula.
- Para alimentar la electrónica y motores, se utilizan baterías LiPo con capacidad de 4000 mAh, tres celdas y velocidad de descarga de 45C.
- El sistema de propulsión del cuadrioptero esta compuesto por cuatro motores 4220-880Kv de 16 polos y cuatro hélices 11x4.5 pulgadas.

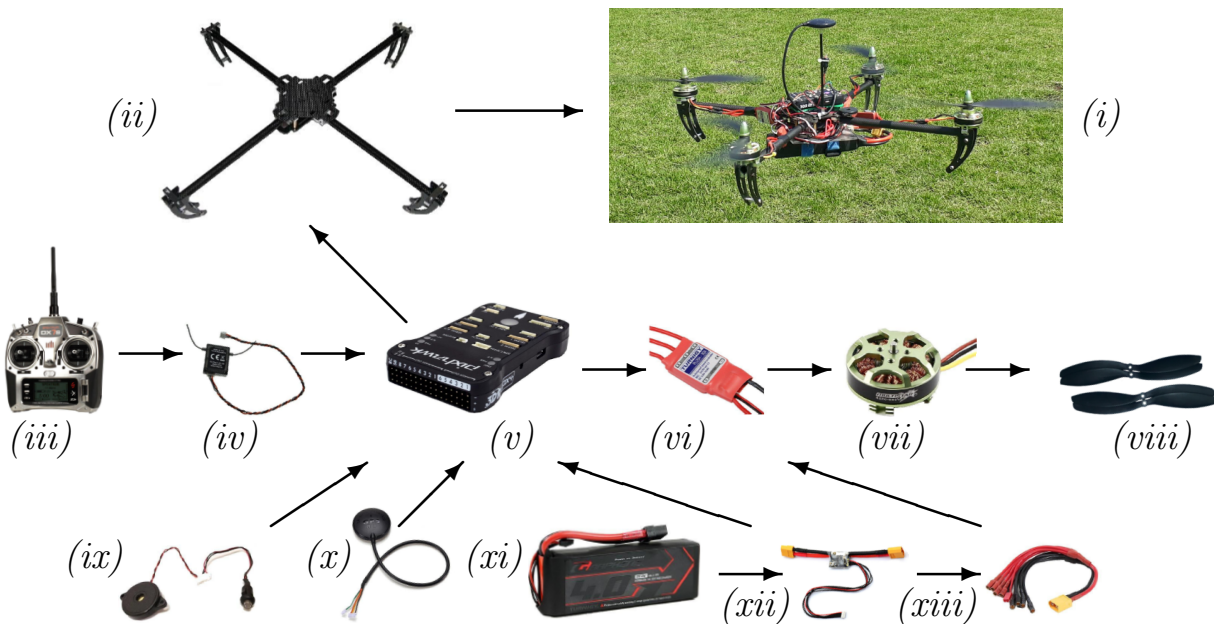


Figura 2.1: Diagrama de conexión de la plataforma experimental. (i) Cuadricóptero, (ii) estructura de fibra de carbono turnigy talon V1, (iii) transmisor DX7s, (iv) receptor DSMX, (v) Pixhawk, (vi) controladores electrónicos de velocidad ( $\times 4$ ), (vii) motores 4220-880Kv de 16 polos ( $\times 4$ ), (viii) hélices 11x4.5 ( $\times 4$ ), (ix) zumbador/interruptor de seguridad, (x) GPS Ublox Neo-M8N, (xi) batería LiPo 4000 mAh 3S, (xii) módulo de potencia, (xiii) arnés de distribución.

## 2.2 Formulación de Euler-Lagrange

Para obtener el modelo dinámico se supone al cuadricóptero como un cuerpo rígido en el espacio, sujeto a una fuerza principal (empuje) y tres momentos (pares).

Un cuerpo rígido en el espacio tridimensional tiene las siguientes coordenadas generalizadas

$$q = \begin{bmatrix} \xi \\ \eta \end{bmatrix} \in \mathbb{R}^6, \quad (2.1)$$

donde  $\xi = [x \ y \ z]^T \in \mathbb{R}^3$  denota el vector de posición del centro de masa del cuadricóptero relativo al marco de referencia inercial  $\mathcal{I}$ , (ver figura 2.2).  $\eta = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$  expresa los ángulos de Euler (que representan la orientación del cuadricóptero) con respecto al marco de referencia inercial,  $\phi$  es el ángulo de alabeo alrededor del eje  $x$ ,  $\theta$  es el ángulo de cabeceo alrededor del eje  $y$  y  $\psi$  es el ángulo de guiñada alrededor del eje  $z$  [23, 24]. Las direcciones positivas de estos ángulos se eligen de acuerdo a la regla de la mano derecha.

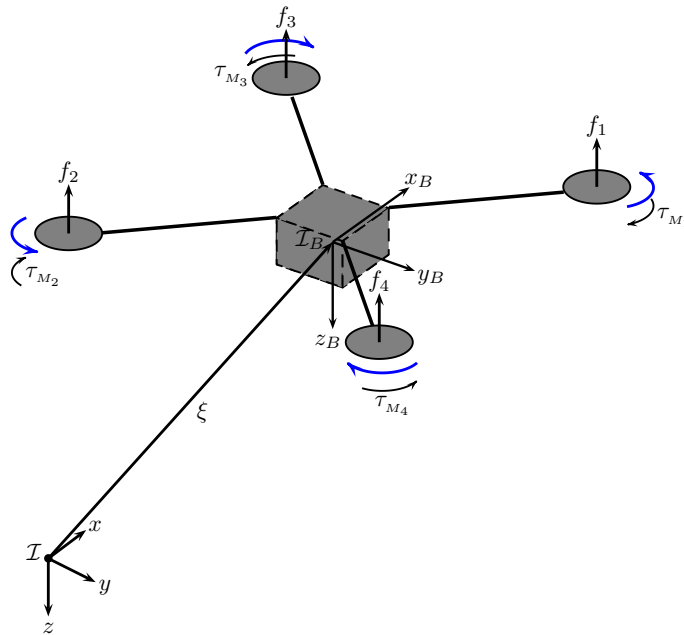


Figura 2.2: Marco inercial  $\mathcal{I}$ , marco fijo al cuerpo  $\mathcal{B}$ , fuerzas y pares.

El Lagrangiano para el cuadricóptero viene dado por

$$\mathcal{L}(q, \dot{q}) = K_{Tras} + K_{Rot} - E_{Pot}, \quad (2.2)$$

donde la energía potencial del cuadricóptero está definida como

$$E_{Pot} = -mgz, \quad (2.3)$$

$m$  es la masa del cuadricóptero,  $g$  es la aceleración debido a la gravedad y  $z$  es la altura del cuadricóptero.

La energía cinética traslacional está definida por

$$K_{Tras} = \frac{1}{2}m\dot{\xi}^T\dot{\xi} = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2), \quad (2.4)$$

donde  $\dot{\xi}$  es el término asociado a la velocidad de traslación del centro de masa del vehículo respecto al marco de referencia fijo.

La energía cinética rotacional es

$$K_{Rot} = \frac{1}{2}\Omega^T I \Omega, \quad (2.5)$$

donde  $I$  es la matriz de inercia y  $\Omega$  es la velocidad angular.

El vector de velocidad angular  $\Omega$  respecto al marco fijo al cuerpo se relaciona con las velocidades generalizadas  $\dot{\eta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$  por medio de la siguiente relación cinemática estándar

$$\Omega = W_{\eta}\dot{\eta}, \quad (2.6)$$

donde

$$W_{\eta} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}. \quad (2.7)$$

Sustituyendo (2.6) en (2.5) se obtiene

$$K_{Rot} = \frac{1}{2}\dot{\eta}^T W_{\eta}^T I W_{\eta} \dot{\eta}. \quad (2.8)$$

## 2.2. Formulación de Euler-Lagrange

---

Definiendo  $\mathbb{J}(\eta) = W_\eta^T I W_\eta$  y sustituyendo en (2.8) se obtiene

$$K_{Rot} = \frac{1}{2} \dot{\eta}^T \mathbb{J}(\eta) \dot{\eta}, \quad (2.9)$$

donde la matriz  $\mathbb{J}(\eta)$  actúa como la matriz de inercia para la energía cinética rotacional total del cuadricóptero, expresada directamente en términos de las coordenadas generalizadas  $\eta$ .

Usando los resultados anteriores, el Lagrangiano se expresa por

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2} m \dot{\xi}^T \dot{\xi} + \frac{1}{2} \dot{\eta}^T \mathbb{J}(\eta) \dot{\eta} + mgz. \quad (2.10)$$

Dado que el Lagrangiano no contiene términos cruzados en la energía cinética que combina  $\dot{\xi}$  con  $\dot{\eta}$ , la expresión (2.10) se reescribe como

$$\mathcal{L}(q, \dot{q}) = \mathcal{L}(\xi, \dot{\xi}) + \mathcal{L}(\eta, \dot{\eta}), \quad (2.11)$$

donde  $\mathcal{L}(\xi, \dot{\xi})$  es el Lagrangiano asociado al movimiento traslacional,  $\mathcal{L}(\eta, \dot{\eta})$  se asocia al movimiento rotacional y se definen como

$$\mathcal{L}(\xi, \dot{\xi}) = \frac{1}{2} m \dot{\xi}^T \dot{\xi} + mgz, \quad (2.12)$$

y

$$\mathcal{L}(\eta, \dot{\eta}) = \frac{1}{2} \dot{\eta}^T \mathbb{J}(\eta) \dot{\eta}. \quad (2.13)$$

El modelo dinámico del cuadricóptero se obtiene de las ecuaciones de Euler-Lagrange con las fuerzas generalizadas externas

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}}{\partial q} = \begin{bmatrix} F_\xi \\ \tau \end{bmatrix}, \quad (2.14)$$

donde  $F_\xi = R \hat{F} \in \mathbb{R}^3$  es la fuerza traslacional aplicada al vehículo debido a la entrada de control principal,  $\tau \in \mathbb{R}^3$  representa los momentos de alabeo (Roll), cabeceo (Pitch) y guiñada (Yaw),  $R$  es la matriz de rotación y representa la orientación del cuadricóptero con respecto al marco inercial  $\mathcal{I}$ .

### Fuerza traslacional y fuerza gravitacional

De la figura 2.2 se sigue que la entrada de control principal o fuerza principal aplicada al vehículo, la cual es generada por los cuatro rotores se expresa como

$$u_z = \sum_{i=1}^4 f_i, \quad (2.15)$$

donde  $f_i$  es la fuerza producida por el motor  $M_i$  para  $i = 1, \dots, 4$ . De manera general,  $f_i = k_i \omega_i^2$  donde  $k_i > 0$  es una constante y  $\omega_i$  es la velocidad angular del  $i$ -ésimo motor.

Se asume que la fuerza (2.15) tiene solo una componente en la dirección  $z$ , de esto sigue que la fuerza total puede escribirse como

$$\hat{F} = \begin{bmatrix} 0 \\ 0 \\ -(f_1 + f_2 + f_3 + f_4) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ u_z \end{bmatrix}. \quad (2.16)$$

El signo negativo se debe al hecho de que el empuje es hacia arriba, mientras que el eje positivo  $z_B$  del marco fijo al cuerpo apunta hacia abajo.

### Pares

Para obtener los momentos o pares generalizados en alabeo y cabeceo, se emplea la siguiente definición

$$\tau_i = r_i \times f_i, \quad (2.17)$$

para  $i = 1, \dots, 4$ .

Considere la figura 2.3, donde  $L_m$  es la distancia del eje de rotación de los rotores y el centro de masa del cuadricóptero.

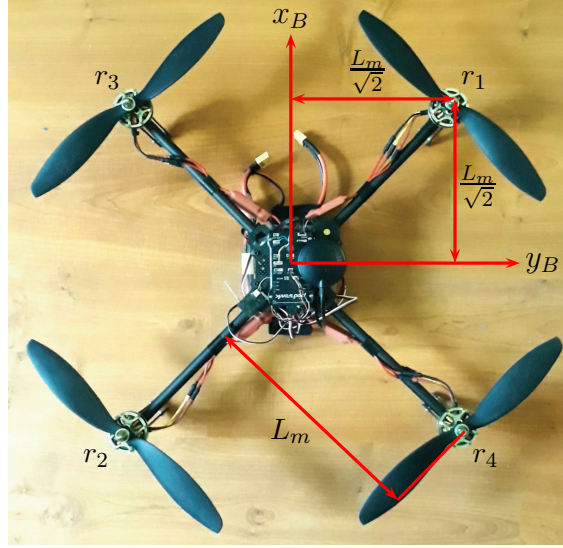


Figura 2.3: Diagrama para obtener los pares.

Para el motor 1 se tiene

$$\tau_1 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{L_m}{\sqrt{2}} & \frac{L_m}{\sqrt{2}} & 0 \\ 0 & 0 & -f_1 \end{vmatrix} = -\frac{L_m}{\sqrt{2}} f_1 \hat{i} + \frac{L_m}{\sqrt{2}} f_1 \hat{j}. \quad (2.18)$$

Para el motor 2 se tiene

$$\tau_2 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ -\frac{L_m}{\sqrt{2}} & -\frac{L_m}{\sqrt{2}} & 0 \\ 0 & 0 & -f_2 \end{vmatrix} = \frac{L_m}{\sqrt{2}} f_2 \hat{i} - \frac{L_m}{\sqrt{2}} f_2 \hat{j}. \quad (2.19)$$

Para el motor 3 se tiene

$$\tau_3 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{L_m}{\sqrt{2}} & -\frac{L_m}{\sqrt{2}} & 0 \\ 0 & 0 & -f_3 \end{vmatrix} = \frac{L_m}{\sqrt{2}} f_3 \hat{i} + \frac{L_m}{\sqrt{2}} f_3 \hat{j}. \quad (2.20)$$



Y para el motor 4 se tiene

$$\tau_4 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ -\frac{L_m}{\sqrt{2}} & \frac{L_m}{\sqrt{2}} & 0 \\ 0 & 0 & -f_4 \end{vmatrix} = -\frac{L_m}{\sqrt{2}}f_4\hat{i} - \frac{L_m}{\sqrt{2}}f_4\hat{j}. \quad (2.21)$$

Para deducir el momento en guiñada, considere la dirección de giro de los motores y la regla de la mano derecha. Los motores 1 y 2 giran en sentido antihorario y los pares que generan son en sentido horario. Los motores 3 y 4 giran en sentido horario y los pares que generan son en sentido antihorario. Usando la regla de la mano derecha se obtiene que los pares generados por los motores 1 y 2 son positivos y los generados por los motores 3 y 4 son negativos.

$$\tau_\psi = \gamma f_1 + \gamma f_2 - \gamma f_3 - \gamma f_4, \quad (2.22)$$

donde  $\gamma = \frac{k_\tau}{k_f}$ .

Por lo tanto, las fuerzas y pares en el cuadricóptero se pueden escribir en forma matricial como

$$\begin{bmatrix} u_z \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ -\frac{L_m}{\sqrt{2}} & \frac{L_m}{\sqrt{2}} & \frac{L_m}{\sqrt{2}} & -\frac{L_m}{\sqrt{2}} \\ \frac{L_m}{\sqrt{2}} & -\frac{L_m}{\sqrt{2}} & \frac{L_m}{\sqrt{2}} & -\frac{L_m}{\sqrt{2}} \\ \gamma & \gamma & -\gamma & -\gamma \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \quad (2.23)$$

### Dinámica traslacional

La ecuación de Euler-Lagrange para el movimiento traslacional es

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \dot{\xi}} \right] - \frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \xi} = F_\xi. \quad (2.24)$$

Realizando las operaciones indicadas se tiene

$$\frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \dot{\xi}} = \frac{\partial}{\partial \dot{\xi}} \left( \frac{1}{2} m \dot{\xi}^T \dot{\xi} + mgz \right) = m \dot{\xi}, \quad (2.25)$$

---

<sup>1</sup>Se tienen  $f_i = k_f \omega_i^2$ , despejando  $\omega_i^2$  y sustituyendo en  $\tau_i = k_\tau \omega_i^2$  se obtiene:  $\tau_i = \frac{k_\tau}{k_f} f_i = \gamma f_i$

## 2.2. Formulación de Euler-Lagrange

---

este resultado se deriva con respecto al tiempo

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \dot{\xi}} \right] = m\ddot{\xi}. \quad (2.26)$$

Por otra parte

$$\frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \xi} = \frac{\partial}{\partial \xi} \left( \frac{1}{2} m \dot{\xi}^T \dot{\xi} + mgz \right) = mgE_z, \quad (2.27)$$

finalmente

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \dot{\xi}} \right] - \frac{\partial \mathcal{L}(\xi, \dot{\xi})}{\partial \xi} = m\ddot{\xi} - mgE_z = F_\xi = R\hat{F}, \quad (2.28)$$

donde  $E_z$  implica que es una fuerza sobre el eje  $z$  inercial,  $R(\phi, \theta, \psi)$  es la matriz de rotación y  $\hat{F}$  es la entrada de control (2.16).

La matriz de rotación<sup>2</sup>  $R(\phi, \theta, \psi)$  tiene la forma (ver [25])

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \quad (2.29)$$

donde

$$c_\gamma \text{ denota } \cos \gamma \text{ y } s_\gamma \text{ denota } \sin \gamma, \quad (\gamma \in \{\phi, \theta, \psi\}). \quad (2.30)$$

De (2.28), (2.29) y recordando que  $F_\xi = R\hat{F}$ , se tiene

$$m \begin{bmatrix} d^2x/dt^2 \\ d^2y/dt^2 \\ d^2z/dt^2 \end{bmatrix} = \begin{bmatrix} u_z(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ u_z(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ u_z(\cos \phi \cos \theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. \quad (2.31)$$

### Dinámica rotacional

Para el movimiento rotacional se tiene la siguiente expresión

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}(\eta, \dot{\eta})}{\partial \dot{\eta}} \right] - \frac{\partial \mathcal{L}(\eta, \dot{\eta})}{\partial \eta} = \tau. \quad (2.32)$$

---

<sup>2</sup> $R = (R_I^B)^T = R_B^I$  es la matriz que permite expresar las fuerzas en el marco inercial.

Realizando las operaciones se tiene

$$\frac{\partial \mathcal{L}(\eta, \dot{\eta})}{\partial \dot{\eta}} = \frac{\partial}{\partial \dot{\eta}} \left( \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta} \right) = \dot{\eta}^T \mathbb{J} \frac{\partial \dot{\eta}}{\partial \dot{\eta}}, \quad (2.33)$$

derivando con respecto al tiempo se llega a

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}(\eta, \dot{\eta})}{\partial \dot{\eta}} \right) = \frac{d}{dt} \left( \dot{\eta}^T \mathbb{J} \frac{\partial \dot{\eta}}{\partial \dot{\eta}} \right) = \mathbb{J} \ddot{\eta} + \dot{\mathbb{J}} \dot{\eta}, \quad (2.34)$$

finalmente, derivando con respecto a  $\eta$

$$\frac{\partial \mathcal{L}(\eta, \dot{\eta})}{\partial \eta} = \frac{\partial}{\partial \eta} \left( \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta} \right). \quad (2.35)$$

Los resultados anteriores se sustituyen en (2.32)

$$\mathbb{J} \ddot{\eta} + \dot{\mathbb{J}} \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}) = \tau. \quad (2.36)$$

Definiendo el vector de Coriolis-centrípetas como

$$\bar{V}(\eta, \dot{\eta}) = \dot{\mathbb{J}} \dot{\eta} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J} \dot{\eta}), \quad (2.37)$$

la expresión (2.36) se escribe como

$$\mathbb{J} \ddot{\eta} + \bar{V}(\eta, \dot{\eta}) = \tau, \quad (2.38)$$

y dado que  $\bar{V}(\eta, \dot{\eta})$  también puede reescribirse como

$$\bar{V}(\eta, \dot{\eta}) = \left( \dot{\mathbb{J}} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J}) \right) \dot{\eta} = C(\eta, \dot{\eta}) \dot{\eta}, \quad (2.39)$$

donde  $C(\eta, \dot{\eta})$  se conoce como el término de Coriolis y contiene los efectos giroscópicos y centrífugos asociados con  $\eta$ .

Sustituyendo (2.39) en (2.38) se obtiene

$$\mathbb{J} \ddot{\eta} = \tau - C(\eta, \dot{\eta}) \dot{\eta}. \quad (2.40)$$

Finalmente el modelo matemático del cuadricóptero está dado por (2.31) y (2.40).

## 2.3 Modelo incremental

### Dinámica traslacional

A fin de compensar la gravedad, se propone la siguiente ley de control

$$u_z = \Delta u_z - mg. \quad (2.41)$$

Sustituyendo (2.41) en (2.31) se obtiene

$$\begin{bmatrix} d^2x/dt^2 \\ d^2y/dt^2 \\ d^2z/dt^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} g + \begin{bmatrix} \cos \phi \operatorname{sen} \theta \cos \psi + \operatorname{sen} \phi \operatorname{sen} \psi \\ \cos \phi \operatorname{sen} \theta \operatorname{sen} \psi - \operatorname{sen} \phi \cos \psi \\ \cos \phi \cos \theta \end{bmatrix} \left( \frac{1}{m} \Delta u_z - g \right). \quad (2.42)$$

Hay que notar que cuando los ángulos son pequeños, el segundo término de la ecuación (2.42) se aproxima de la siguiente manera

$$\begin{bmatrix} \cos \phi \operatorname{sen} \theta \cos \psi + \operatorname{sen} \phi \operatorname{sen} \psi \\ \cos \phi \operatorname{sen} \theta \operatorname{sen} \psi - \operatorname{sen} \phi \cos \psi \\ \cos \phi \cos \theta \end{bmatrix} \approx \begin{bmatrix} \theta \\ -\phi \\ 1 \end{bmatrix}. \quad (2.43)$$

Sumando y restando (2.43) en (2.42), se obtiene la siguiente expresión (recordar nomenclatura (2.30))

$$\begin{aligned} \begin{bmatrix} d^2x/dt^2 \\ d^2y/dt^2 \\ d^2z/dt^2 \end{bmatrix} &= \begin{bmatrix} -\theta g \\ \phi g \\ \frac{1}{m} \Delta u_z \end{bmatrix} + \begin{bmatrix} \theta - (c_\phi s_\theta c_\psi + s_\phi s_\psi) \\ -\phi - (c_\phi s_\theta s_\psi - s_\phi c_\psi) \\ 1 - c_\phi c_\theta \end{bmatrix} g + \begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta - 1 \end{bmatrix} \frac{1}{m} \Delta u_z \\ &= \begin{bmatrix} -\theta g \\ \phi g \\ \frac{1}{m} \Delta u_z \end{bmatrix} + \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \end{aligned} \quad (2.44)$$

donde

$$\begin{aligned} q_x &= \theta g - q_{xx}g + \frac{1}{m} \Delta u_z q_{xx}, \\ q_y &= -\phi g - q_{yy}g + \frac{1}{m} \Delta u_z q_{yy}, \\ q_z &= -q_{zz}g + \frac{1}{m} \Delta u_z q_{zz}, \end{aligned} \quad (2.45)$$

y

$$\begin{aligned}
 q_{xx} &= \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi, \\
 q_{yy} &= \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi, \\
 q_{zz} &= \cos \phi \cos \theta - 1.
 \end{aligned} \tag{2.46}$$

donde  $q_x$ ,  $q_y$  y  $q_z$  describen la parte no lineal de la dinámica traslacional.

Note que para pequeños ángulos:  $q_x \rightarrow 0$ ,  $q_y \rightarrow 0$  y  $q_z \rightarrow 0$ .

### Dinámica rotacional

Recordando que  $\mathbb{J}(\eta) = W_\eta^T I W_\eta$  y despejando  $\ddot{\eta}$  de (2.40) se obtiene

$$\ddot{\eta} = \mathbb{J}^{-1}(\eta)\tau - \mathbb{J}^{-1}(\eta)C(\eta, \dot{\eta})\dot{\eta}, \tag{2.47}$$

donde  $\mathbb{J}^{-1}(\eta) = W_\eta^{-1}I^{-1}(W_\eta^T)^{-1}$ ,  $I$  es la matriz de inercia del cuadricóptero dada por

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \tag{2.48}$$

considerando  $\phi = \theta = 0$ , la matriz  $\mathbb{J}^{-1}(\eta)$  se expresa como

$$\mathbb{J}^{-1}(0) = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}. \tag{2.49}$$

Sumando  $\pm \mathbb{J}^{-1}(0)\tau$  en la ecuación (2.47) y agrupando términos se obtiene

$$\ddot{\eta} = \mathbb{J}^{-1}(0)\tau + q_{rot}, \tag{2.50}$$

donde

$$q_{rot} = \begin{bmatrix} q_\phi \\ q_\theta \\ q_\psi \end{bmatrix} = (\mathbb{J}^{-1}(\eta) - \mathbb{J}^{-1}(0))\tau - \mathbb{J}^{-1}(\eta)C(\eta, \dot{\eta})\dot{\eta}. \tag{2.51}$$

## 2.4. Representaciones de estado

---

Así, la expresión (2.50) se puede reescribir como

$$\begin{bmatrix} d^2\phi/dt^2 \\ d^2\theta/dt^2 \\ d^2\psi/dt^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} u_y \\ u_x \\ u_\psi \end{bmatrix} + \begin{bmatrix} q_\phi \\ q_\theta \\ q_\psi \end{bmatrix}, \quad (2.52)$$

donde  $q_\phi$ ,  $q_\theta$  y  $q_\psi$  describen la parte no lineal de la dinámica rotacional y  $u_y = \tau_\phi$ ,  $u_x = \tau_\theta$  y  $u_\psi = \tau_\psi$ .

Finalmente se tiene

$$\begin{aligned} d^2x/dt^2 &= -\theta g + q_x, \\ d^2y/dt^2 &= \phi g + q_y, \end{aligned} \quad (2.53)$$

$$d^2z/dt^2 = \Delta u_z/m + q_z,$$

$$d^2\phi/dt^2 = u_y/I_{xx} + q_\phi,$$

$$d^2\theta/dt^2 = u_x/I_{yy} + q_\theta, \quad (2.54)$$

$$d^2\psi/dt^2 = u_\psi/I_{zz} + q_\psi,$$

## 2.4 Representaciones de estado

De las ecuaciones (2.53) y (2.54) se derivan las siguientes representaciones de estado para las dinámicas en  $x$  (subsistema  $x - \theta$ ), en  $y$  (subsistema  $y - \phi$ ), en guiñada (subsistema  $\psi$ ) y en  $z$  (subsistema  $z$ ).

**Representación de estado del subsistema  $x - \theta$**

$$\frac{d}{dt}x_x = A_x x_x + B_x u_x + S_x q_{ox}, \quad (2.55)$$

$$x = C_x x_x,$$

$$A_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{I_{yy}} \end{bmatrix}, \quad S_x = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T, \quad (2.56)$$

donde  $x_x = \left[ x \quad \frac{d}{dt}x \quad \theta \quad \frac{d}{dt}\theta \right]^T$ ,  $q_{ox} = \left[ q_x \quad q_\theta \right]^T$ .

### Representación de estado del subsistema $y - \phi$

$$\begin{aligned} \frac{d}{dt}x_y &= A_y x_y + B_y u_y + S_y q_{oy}, \\ y &= C_y x_y, \end{aligned} \quad (2.57)$$

$$A_y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{I_{xx}} \end{bmatrix}, \quad S_y = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T, \quad (2.58)$$

donde  $x_y = \left[ y \quad \frac{d}{dt}y \quad \phi \quad \frac{d}{dt}\phi \right]^T$ ,  $q_{oy} = \left[ q_y \quad q_\phi \right]^T$ .

### Representación de estado del subsistema $\psi$

$$\begin{aligned} \frac{d}{dt}x_\psi &= A_\psi x_\psi + B_\psi u_\psi + S_\psi q_\psi, \\ \psi &= C_\psi x_\psi, \end{aligned} \quad (2.59)$$

$$A_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_\psi = \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \end{bmatrix}, \quad S_\psi = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_\psi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \quad (2.60)$$

donde  $x_\psi = \left[ \psi \quad \frac{d}{dt}\psi \right]^T$ .

### Representación de estado del subsistema $z$

$$\begin{aligned} \frac{d}{dt}x_z &= A_z x_z + B_z \Delta u_z + S_z q_z, \\ z &= C_z x_z, \end{aligned} \quad (2.61)$$

$$A_z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_z = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad S_z = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_z = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T, \quad (2.62)$$

donde  $x_z = \left[ z \quad \frac{d}{dt}z \right]^T$ .

## 2.5 Conclusiones

En este capítulo aplicó el enfoque de Euler-Lagrange para obtener un modelo matemático del cuadricóptero. El modelo subactuado que se obtuvo conserva las principales características que deben tenerse en cuenta en el diseño de las leyes de control. La configuración analizada es la que dispone los motores en “X”, dos motores frontales y dos motores traseros.

A partir de las ecuaciones matemáticas del cuadricóptero se obtuvo el modelo incremental, en el que se identifican las señales que describen la parte no lineal de la dinámica traslacional y rotacional. Del modelo incremental se obtuvieron cuatro representaciones de estado, estas se usarán en la secciones siguientes para diseñar leyes de control enfocadas en la estabilización en orientación y posición del cuadricóptero.





# Capítulo 3

## Linealización

En este capítulo primeramente se obtiene un modelo lineal aproximado del cuadricóptero a partir del modelo incremental desarrollado en el capítulo anterior. Posteriormente se presenta un método de linealización que se basa en la estructura interna de la descripción de estado del sistema linealizado, caracterizado por el par  $(A, B)$ . Se presentan las condiciones necesarias que se deben cumplir para llevar a cabo este enfoque de linealización. A continuación se introduce un cambio de variable que permite obtener una nueva señal de incertidumbre no lineal  $q_*$  a partir de la señal de incertidumbre no lineal  $q$ . Se aplica una realimentación de estado a cada uno de los subsistemas del cuadricóptero y posteriormente se desarrolla el método de linealización.

### 3.1 Linealización aproximada

Para obtener un modelo lineal a partir de modelo incremental, se consideran los siguientes puntos

1. Los ángulos de alabeo, cabeceo y guiñada son pequeños, esto permite hacer las siguientes aproximaciones:  $\cos \phi \approx 1$ ,  $\cos \theta \approx 1$ ,  $\cos \psi \approx 1$ ,  $\sin \phi \approx \phi$ ,  $\sin \theta \approx \theta$  y  $\sin \psi \approx \psi$ .
2. Los productos de dos o más variables  $(\phi\theta)$ , productos de derivadas de ángulos  $(\dot{\theta}\dot{\phi})$ ,

productos de un ángulo y una entrada de control ( $\dot{\phi}u_x$ ) que se obtienen después de aplicar el punto anterior son pequeños y pueden despreciarse.

Bajo los puntos anteriores, las señales que describen la parte no lineal (2.45) y (2.46) de la dinámica traslacional tienen la siguiente forma

$$\begin{aligned} q_x &= \theta g - g (\cos \phi \sen \theta \cos \psi + \sen \phi \sen \psi) + \frac{1}{m} \Delta u_z (\cos \phi \sen \theta \cos \psi + \sen \phi \sen \psi) \\ &= \theta g - g(\theta + \phi\psi) + \frac{1}{m} \Delta u_z (\theta + \phi\psi) \\ &= 0. \end{aligned} \quad (3.1)$$

$$\begin{aligned} q_y &= -\phi g - g (\cos \phi \sen \theta \sen \psi - \sen \phi \cos \psi) + \frac{1}{m} \Delta u_z (\cos \phi \sen \theta \sen \psi - \sen \phi \cos \psi) \\ &= -\phi g - g(\theta\psi - \phi) + \frac{1}{m} \Delta u_z (\theta\psi - \phi) \\ &= 0, \end{aligned} \quad (3.2)$$

$$\begin{aligned} q_z &= -(\cos \phi \cos \theta - 1) g + \frac{1}{m} \Delta u_z (\cos \phi \cos \theta - 1) \\ &= -(1 - 1)g + \frac{1}{m} \Delta u_z (1 - 1) \\ &= 0 \end{aligned} \quad (3.3)$$

Con respecto a las señales que describen la parte no lineal de la dinámica rotacional (2.51), se tiene

$$\mathbb{J}(\eta) = \begin{bmatrix} I_{xx} & 0 & -I_{xx}s\theta \\ 0 & I_{yy}c_\phi^2 + I_{zz}s_\phi^2 & (I_{yy} - I_{zz})c_\phi s_\phi c_\theta \\ -I_{xx}s\theta & (I_{yy} - I_{zz})c_\phi s_\phi c_\theta & I_{xx}s_\theta^2 + (I_{yy}s_\phi^2 + I_{zz}c_\phi^2)c_\theta^2 \end{bmatrix}, \quad (3.4)$$

y

$$C(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \quad (3.5)$$

### 3.1. Linealización aproximada

---

donde

$$\begin{aligned}
c_{11} &= 0 \\
c_{12} &= -I_{xx}c_\theta\dot{\psi} - (I_{zz} - I_{yy}) \left( c_\phi s_\phi \dot{\theta} + (s_\phi^2 - c_\phi^2)c_\theta\dot{\psi} \right) \\
c_{13} &= (I_{zz} - I_{yy})c_\phi s_\phi c_\theta^2 \dot{\psi} \\
c_{21} &= I_{xx}c_\theta\dot{\psi} + (I_{zz} - I_{yy}) \left( c_\phi s_\phi \dot{\theta} + (s_\phi^2 - c_\phi^2)c_\theta\dot{\psi} \right) \\
c_{22} &= (I_{zz} - I_{yy})c_\phi s_\phi \dot{\phi} \\
c_{23} &= (I_{zz}c_\phi^2 + I_{yy}s_\phi^2 - I_{xx}) c_\theta s_\theta \dot{\psi} \\
c_{31} &= -I_{xx}c_\theta\dot{\theta} - (I_{zz} - I_{yy})c_\phi s_\phi c_\theta^2 \dot{\psi} \\
c_{32} &= I_{xx}c_\theta s_\theta \dot{\psi} + I_{yy} \left( (c_\phi^2 - s_\phi^2)c_\theta\dot{\phi} - c_\phi s_\phi s_\theta \dot{\theta} - s_\phi^2 c_\theta s_\theta \dot{\psi} \right) \\
&\quad - I_{zz} \left( (c_\phi^2 - s_\phi^2)c_\theta\dot{\phi} - c_\phi s_\phi s_\theta \dot{\theta} + c_\phi^2 c_\theta s_\theta \dot{\psi} \right) \\
c_{33} &= I_{xx}c_\theta s_\theta \dot{\theta} + I_{yy} \left( c_\phi s_\phi c_\theta \dot{\phi} - s_\phi^2 s_\theta \dot{\theta} \right) c_\theta - I_{zz} \left( s_\phi c_\theta^2 \dot{\phi} + c_\phi c_\theta s_\theta \dot{\theta} \right) c_\phi
\end{aligned}$$

Al tomar en cuenta ángulos pequeños, las variables  $q_\phi$ ,  $q_\theta$  y  $q_\psi$  de la expresión (2.52) se aproximan a cero (ver (2.51), (3.4), (3.5)), es decir

$$q_\phi = 0. \quad (3.6)$$

$$q_\theta = 0. \quad (3.7)$$

$$q_\psi = 0. \quad (3.8)$$

Considerando (3.1)-(3.8), las representaciones de estado (2.55), (2.57), (2.59) y (2.61) para ángulos pequeños se aproximan de la siguiente manera

$$\frac{d}{dt}x_x = A_x x_x + B_x u_x, \quad (3.9)$$

$$x = C_x x_x.$$

$$\frac{d}{dt}x_y = A_y x_y + B_y u_y, \quad (3.10)$$

$$y = C_y x_y.$$

$$\begin{aligned}\frac{d}{dt}x_\psi &= A_\psi x_\psi + B_\psi u_\psi, \\ \psi &= C_\psi x_\psi.\end{aligned}\tag{3.11}$$

$$\begin{aligned}\frac{d}{dt}x_z &= A_z x_z + B_z \Delta u_z, \\ z &= C_z x_z.\end{aligned}\tag{3.12}$$

## 3.2 Linealización exacta entrada-salida

Ahora introducimos una nueva técnica de linealización para la siguiente clase de sistemas no lineales.

$$\begin{aligned}\frac{d}{dt}x(t) &= f(x(t)) + g(x(t))u(t), \\ y(t) &= Cx(t),\end{aligned}\tag{3.13}$$

donde  $x \in \mathbb{R}^n$  es la variable de estado,  $y \in \mathbb{R}^p$  es la variable de salida,  $u \in \mathbb{R}^m$  es la entrada de control. Aquí  $p \leq m$ ,  $f(x)$  y  $g(x)$  son campos vectoriales analíticos en  $\mathbb{R}^n$  y  $C \in \mathbb{R}^{p \times n}$  es una matriz inyectiva, esto es,  $\ker C^T = 0$ .

La metodología de linealización propuesta se basa en la estructura interna dada de la descripción de estado del sistema linealizado caracterizado por el par  $(A, B)$  con

$$A \doteq \left[ \frac{\partial f}{\partial x} \right]_{x=0} \in \mathbb{R}^{n \times n}, \quad B \doteq [g]_{x=0} \in \mathbb{R}^{n \times m},\tag{3.14}$$

Dada la caracterización (3.14), (3.13) se puede reescribir de la siguiente manera

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + f_o(x(t), u(t)), \\ y(t) &= Cx(t)\end{aligned}\tag{3.15}$$

donde

$$\begin{aligned}f_o(x(t), u(t)) &= \Delta f(x(t)) + \Delta g(x(t))u(t) \\ \Delta f(x(t)) &= f(x(t)) - Ax(t) \\ \Delta g(x(t)) &= g(x(t)) - B\end{aligned}\tag{3.16}$$

Note que en la función  $f_o(x(t), u(t))$  están contenidas las no linealidades del sistema, las cuales tienden a cero alrededor del punto de equilibrio.

Ahora se hacen las siguientes hipótesis.

**H1.** *Suposiciones estructurales:*

1.  $\ker B = 0$ .
2. El par  $(A, B)$  es controlable, esto es [26] :  $\mathbb{R}^n = \text{Im } B + A \text{Im } B + \cdots + A^{n-1} \text{Im } B$ .
3. El par  $(C, A)$  es observable.

### 3.2.1 Descomposición estructural

**H2.** *Existe un vector de incertidumbre  $q \in \mathbb{R}^\mu$  y una transformación lineal  $S : \mathbb{R}^\mu \rightarrow \mathbb{R}^n$  tal que*

$$f_o(x(t), u(t)) = Sq(x(t), u(t)). \quad (3.17)$$

Bajo la hipótesis **H2**, (3.15) toma la siguiente forma

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + Bu(t) + Sq(x(t), u(t)), \\ y(t) &= Cx(t). \end{aligned} \quad (3.18)$$

$q \in \mathbb{R}^\mu$  es una variable en donde se han concentrado todas las no linealidades del sistema. A esta variable la llamaremos *señal de perturbación no lineal*.

La *señal de perturbación no lineal*,  $q$ , tiende a cero cerca del punto de equilibrio.

**Lema 3.2.1.** *Asumiendo que **H1.1-H1.2** son ciertas. Entonces*

1. *Existen transformaciones lineales  $M : \mathbb{R}^n \rightarrow \mathbb{R}^n$  y  $X : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , tales que*

$$AM + BX = I \quad (3.19)$$

2. *Las transformaciones lineales  $X$  y  $M$  resuelven (3.19) y satisfacen las siguientes condiciones*

$$X = B^\ell(I - AM), \quad (3.20)$$

$$\widehat{N}AM = \widehat{N}, \quad (3.21)$$

donde  $\widehat{N} : \mathbb{R}^n \rightarrow \mathbb{R}^n / \text{Im } B$  es la proyección canónica<sup>1</sup> y  $B^\ell : \mathbb{R}^n \rightarrow \mathbb{R}^m$  es la inversa por la izquierda de  $B : \mathbb{R}^m \rightarrow \mathbb{R}^n$ , es decir  $B^\ell B = I$ .

3. Existe una solución nilpotente  $M : \mathbb{R}^n \rightarrow \mathbb{R}^n$  de (3.21), con un índice de nilpotencia no mayor que  $n$ , que satisface

$$M^n = 0 \quad (3.22)$$

*Prueba del lema 3.2.1:* Dado que (ver **H1.2**)  $\mathbb{R}^n = \text{Im } B + A \text{Im } B + \dots + A^{n-1} \text{Im } B \subset \text{Im } A + \text{Im } B \subset \mathbb{R}^n$  entonces  $\text{Im } A + \text{Im } B = \mathbb{R}^n$ . Lo cual implica la existencia de las matrices  $M$  y  $X$  satisfaciendo (3.19). Ya que  $B^\ell B = I$ , adicionalmente obtenemos (3.20).

Usando el teorema de Brunovsky [27], uno puede deducir que existen bases en  $\mathbb{R}^n$ ,  $\mathbb{R}^m$ , y  $\mathbb{R}^n / \text{Im } B$  tal que

$$\widehat{N} A = \text{BD}\{G_1, \dots, G_m\} \text{ y } \widehat{N} = \text{BD}\{N_1, \dots, N_m\}. \quad (3.23)$$

Aquí  $G_i$  y  $N_i$  son matrices en  $\mathbb{R}^{\kappa_{i-1} \times \kappa_i}$ ,  $i \in \{1, \dots, m\}$  con la propiedad  $\sum_{i=1}^m \kappa_i = n$  y

$$G_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \text{ y } N_i = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}. \quad (3.24)$$

La última relación implica que la matriz  $M$  está definida como sigue<sup>2</sup>

$$M = \text{BD}\{M_{1[\kappa_1]}, \dots, M_{m[\kappa_m]}\}, M_{i[\kappa_i]} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{\kappa_i \times \kappa_i}, \quad (3.25)$$

donde  $i \in \{1, \dots, m\}$ . Satisface las condiciones (3.21) y (3.22). La prueba está completa.  $\square$

<sup>1</sup> $\widehat{N}$  es una matriz suryectiva tal que  $\ker \widehat{N} = \text{Im } B$ .

<sup>2</sup>Notemos que esta matriz  $M$  corresponde a la matriz transpuesta de la forma canónica de Brunovsky de  $A$ .

### 3.2.2 El esquema exacto de la linealización por realimentación estructural

Definiendo la matriz de controlabilidad

$$\mathcal{C}_{(M,S)} = \begin{bmatrix} S & MS & \dots & M^{(n-1)}S \end{bmatrix}, \quad (3.26)$$

el operador diferencial

$$\Psi_n(d/dt) = \begin{bmatrix} I & Id/dt & \dots & Id^{n-1}/dt^{n-1} \end{bmatrix}^T, \quad (3.27)$$

y haciendo la siguiente hipótesis adicional

**H3.** *El subespacio  $M \operatorname{Im} S$  está contenido en el subespacio inobservable  $\langle \ker C \mid M \rangle = \ker C \cap M^{-1} \ker C \cap \dots \cap M^{-(n-1)} \ker C$ , es decir*

$$CM\mathcal{C}_{(M,S)} = 0. \quad (3.28)$$

se tiene el siguiente teorema

**Teorema 3.2.1.** *Dadas las transformaciones lineales  $M$  y  $X$  del lema 3.2.1, consideremos el siguiente cambio de variable*

$$\zeta(t) = x(t) + M\mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)). \quad (3.29)$$

*Bajo las suposiciones de **H1.1-H1.2** y **H3**, la representación de estado (3.18) es externamente equivalente a*<sup>3</sup>

$$\begin{aligned} \frac{d}{dt}\zeta(t) &= A\zeta(t) + B(u(t) + q_*(x(t), u(t))), \\ y(t) &= C\zeta(t), \end{aligned} \quad (3.30)$$

*donde la señal de incertidumbre no lineal,  $q_*$ , está definida como*

$$q_*(x(t), u(t)) = X\mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)). \quad (3.31)$$

---

<sup>3</sup>Recordar que dos representaciones se llaman externamente equivalentes si los correspondientes conjuntos de todas las posibles trayectorias para las variables externas expresadas en una partición entrada/salida  $(u, y)$  son la misma [28, 29].



*Prueba del teorema 3.2.1:* Dada una condición inicial  $x(0) = x_0 \in \mathbb{R}^n$  y una entrada de control  $u(t) \in \mathbb{R}^m$ ,  $t \geq 0$ , la solución de (3.18) es

$$\begin{aligned} x(t) &= \exp(At)x_0 + \int_0^t \exp(A(t-\tau)) (Bu(\tau) + Sq(x(\tau), u(\tau))) d\tau, \\ y(t) &= Cx(t). \end{aligned} \quad (3.32)$$

de (3.29) y (3.28) deducimos

$$C\zeta(t) = Cx(t) + CM\mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)) = Cx(t). \quad (3.33)$$

De (3.29), (3.26), (3.27), (3.18), (3.21) y (3.22) se deduce (recordar que  $\widehat{N}B = 0$ )

$$\begin{aligned} \widehat{N} \left( \mathbf{I} \frac{d}{dt} - A \right) \zeta(t) &= \widehat{N} \left( \mathbf{I} \frac{d}{dt} - A \right) x(t) \\ &\quad + \widehat{N} \left( \mathbf{I} \frac{d}{dt} - A \right) M \left( S + \sum_{i=1}^{n-1} M^i S \frac{d^i}{dt^i} \right) q(x(t), u(t)) \\ &= \widehat{N} S q(x(t), u(t)) \\ &\quad + \left( \widehat{N} M \frac{d}{dt} - \widehat{N} \right) \left( \mathbf{I} + \sum_{i=1}^{n-1} M^i \frac{d^i}{dt^i} \right) S q(x(t), u(t)) \\ &= \widehat{N} S q(x(t), u(t)) + \left( \widehat{N} M^n \frac{d^n}{dt^n} - \widehat{N} \right) S q(x(t), u(t)) \\ &= 0. \end{aligned} \quad (3.34)$$

De (3.29), (3.26), (3.27), (3.18), (3.19), (3.22) y (3.31) se deduce (recordar que  $B^\ell B = \mathbf{I}$ )

$$\begin{aligned} B^\ell \left( \mathbf{I} \frac{d}{dt} - A \right) \zeta(t) &= B^\ell \left( \mathbf{I} \frac{d}{dt} - A \right) x(t) \\ &\quad + B^\ell \left( \mathbf{I} \frac{d}{dt} - A \right) M \left( S + \sum_{i=1}^{n-1} M^i S \frac{d^i}{dt^i} \right) q(x(t), u(t)) \\ &= u(t) + B^\ell S q(x(t), u(t)) \\ &\quad + \left( B^\ell M \frac{d}{dt} - B^\ell + X \right) \left( \mathbf{I} + \sum_{i=1}^{n-1} M^i \frac{d^i}{dt^i} \right) S q(x(t), u(t)) \\ &= u(t) + B^\ell S q(x(t), u(t)) \\ &\quad + \left( B^\ell M^n \frac{d^n}{dt^n} - B^\ell \right) S q(x(t), u(t)) \\ &\quad + X \left( \mathbf{I} + \sum_{i=1}^{n-1} M^i \frac{d^i}{dt^i} \right) S q(x(t), u(t)) \\ &= u(t) + X \mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)) \\ &= u(t) + q_*(x(t), u(t)). \end{aligned} \quad (3.35)$$

La prueba está completa. □

La importancia del teorema 3.2.1 es que proporciona el cambio de base (3.29) con el cual al vector de incertidumbre (señal de perturbación no lineal),  $q$ , se le mapea a una nueva señal de incertidumbre no lineal,  $q_*$ , que se encuentra dentro de la imagen de la matriz  $B$ , por lo que la ley de control

$$u(t) = -q_*(x(t), u(t)), \quad (3.36)$$

linealiza exactamente a (3.30). La función de transferencia de (3.30) es

$$F_\zeta(s) = C(sI - A)^{-1}B. \quad (3.37)$$

## 3.3 Linealización exacta entrada-salida del cuadricóptero

### 3.3.1 Realimentaciones de estado

Primero se aplicará una realimentación de estado a los subsistemas (2.55)-(2.62) como se muestra a continuación.

#### Representación de estado del subsistema $x - \theta$

A la representación de estado (2.55) se le aplica una realimentación de estado de la forma

$$u_x = F_x x_x + \bar{u}_x, \quad (3.38)$$

donde  $F_x = \begin{bmatrix} a_{x,4} I_{yy}/g & a_{x,3} I_{yy}/g & -a_{x,2} I_{yy} & -a_{x,1} I_{yy} \end{bmatrix}$ .

La descripción en espacio de estado del sistema en lazo cerrado que se expresa como

$$\begin{aligned} \frac{d}{dt} x_x &= A_{F_x} x_x + B_x \bar{u}_x + S_x q_{ox}, \\ x &= C_x x_x, \end{aligned} \quad (3.39)$$

donde

$$A_{F_x} = A_x + B_x F_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ a_{x,4}/g & a_{x,3}/g & -a_{x,2} & -a_{x,1} \end{bmatrix}, \quad B_x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{I_{yy}} \end{bmatrix}, \quad S_x = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_x = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$
(3.40)

Note que el polinomio característico de la matriz en lazo cerrado  $A_{F_x}$  dado por

$$\pi_{x_x}(s) = \det(sI - A_{F_x}) = s^4 + a_{x,1}s^3 + a_{x,2}s^2 + a_{x,3}s + a_{x,4}, \quad (3.41)$$

es un polinomio Hurwitz.

### Representación de estado del subsistema $y - \phi$

De manera similar al caso anterior, se propone la siguiente realimentación de estado

$$u_y = F_y x_y + \bar{u}_y, \quad (3.42)$$

donde  $F_y = \begin{bmatrix} -a_{y,4}I_{xx}/g & -a_{y,3}I_{xx}/g & -a_{y,2}I_{xx} & -a_{y,1}I_{xx} \end{bmatrix}$ .

Después de aplicar (3.42) a (2.57), se obtiene el sistema en lazo cerrado que se muestra a continuación

$$\frac{d}{dt}x_y = A_{F_y}x_y + B_y\bar{u}_y + S_y q_{oy}, \quad (3.43)$$

$$y = C_y x_y,$$

donde

$$A_{F_y} = A_y + B_y F_y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ -a_{y,4}/g & -a_{y,3}/g & -a_{y,2} & -a_{y,1} \end{bmatrix}, \quad B_y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{I_{xx}} \end{bmatrix}, \quad S_y = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$
(3.44)

### 3.3. Linealización exacta entrada-salida del cuadricóptero

---

El polinomio característico de la matriz en lazo cerrado  $A_{F_y}$  dado por

$$\pi_{x_y}(s) = \det(sI - A_{F_y}) = s^4 + a_{y,1}s^3 + a_{y,2}s^2 + a_{y,3}s + a_{y,4}, \quad (3.45)$$

es un polinomio Hurwitz.

#### Representación de estado del subsistema $\psi$

A la representación de estado (2.59) se le aplica una realimentación de estado de la forma

$$u_\psi = F_\psi x_\psi + \bar{u}_\psi, \quad (3.46)$$

donde  $F_\psi = \begin{bmatrix} -a_{\psi,2}I_{zz} & -a_{\psi,1}I_{zz} \end{bmatrix}$ .

La descripción en espacio de estado del sistema en lazo cerrado está dada por

$$\begin{aligned} \frac{d}{dt}x_\psi &= A_{F_\psi}x_\psi + B_\psi\bar{u}_\psi + S_\psi q_\psi, \\ \psi &= C_\psi x_\psi, \end{aligned} \quad (3.47)$$

donde

$$\begin{aligned} A_{F_\psi} &= A_\psi + B_\psi F_\psi = \begin{bmatrix} 0 & 1 \\ -a_{\psi,2} & -a_{\psi,1} \end{bmatrix}, & B_\psi &= \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \end{bmatrix}, & S_\psi &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ C_\psi &= \begin{bmatrix} 1 & 0 \end{bmatrix}. \end{aligned} \quad (3.48)$$

El polinomio característico de la matriz en lazo cerrado  $A_{F_\psi}$  expresado como

$$\pi_{x_\psi}(s) = \det(sI - A_{F_\psi}) = s^2 + a_{\psi,1}s + a_{\psi,2}, \quad (3.49)$$

es un polinomio Hurwitz.

#### Representación de estado del subsistema $z$

Se propone la siguiente realimentación de estado

$$\Delta u_z = F_z x_z + \bar{u}_z, \quad (3.50)$$

donde  $F_z = \begin{bmatrix} -a_{z,2}m & -a_{z,1}m \end{bmatrix}$ . Después de aplicar (3.50) a (2.61), se obtiene el siguiente sistema en lazo cerrado

$$\begin{aligned} \frac{d}{dt}x_z &= A_{F_z}x_z + B_z\bar{u}_z + S_zq_z, \\ z &= C_zx_z, \end{aligned} \quad (3.51)$$

donde

$$\begin{aligned} A_{F_z} &= A_z + B_zF_z = \begin{bmatrix} 0 & 1 \\ -a_{z,2} & -a_{z,1} \end{bmatrix}, & B_z &= \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, & S_z &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ C_z &= \begin{bmatrix} 1 & 0 \end{bmatrix}. \end{aligned} \quad (3.52)$$

El polinomio característico de la matriz en lazo cerrado  $A_{F_z}$  dado por

$$\pi_{x_z}(s) = \det(sI - A_{F_z}) = s^2 + a_{z,1}s + a_{z,2}, \quad (3.53)$$

es un polinomio Hurwitz.

### 3.3.2 Descomposición estructural

Las representaciones de estado (3.47) y (3.51) ya se encuentran en la forma (3.30) del Teorema 3.2.1, es decir  $\text{Im } S_\psi \subset \text{Im } B_\psi$ , por lo que  $q_\psi$  puede ser anulada directamente con  $\bar{u}_\psi$  (lo mismo para el subsistema  $z$ ). Por lo que solamente consideramos (3.39) y (3.43).

#### Descomposición estructural del subsistema $x - \theta$

Observemos que

- $\ker B_x = 0$ .
- el par  $(A_{F_x}, B_x)$  es controlable.
- el par  $(C_x, A_{F_x})$  es observable.

Ahora consideramos la solución de la ecuación algebraica  $A_{F_x}M_x + B_xX_x = I_4$  (ver (3.19)),

### 3.3. Linealización exacta entrada-salida del cuadricóptero

---

(3.25) y (3.20)) que puede expresarse como

$$M_x = T_x M_{1[4]} T_x^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1/g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_x^\ell = \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_{yy} \end{bmatrix}^T, \quad (3.54)$$

$$T_x = (1/I_{yy}) \begin{bmatrix} -g & 0 & 0 & 0 \\ 0 & -g & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad X_x = B_x^\ell (I_4 - A_{F_x} M_x) = I_{yy} \begin{bmatrix} -a_{x,3}/g \\ -a_{x,2}/g \\ a_{x,1} \\ 1 \end{bmatrix}^T.$$

De (3.54) y (3.40) se obtiene<sup>4</sup>

$$A_{F_x} M_x + B_x X_x = I_4, \quad M_x^4 = 0, \quad C_x M_x \mathcal{C}_{(M_x, S_x)} = 0,$$

$$M_x \mathcal{C}_{(M_x, S_x)} \Psi_4(d/dt) = - \left( \frac{1}{g} \right) \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ d/dt & 0 \end{bmatrix}, \quad (3.55)$$

$$X_x \mathcal{C}_{(M_x, S_x)} \Psi_4\left(\frac{d}{dt}\right) = I_{yy} \left[ -\frac{1}{g} \left( a_{x,2} + a_{x,1} \frac{d}{dt} + \frac{d^2}{dt^2} \right) \quad 1 \right].$$

Además, de (3.55), (3.29) y (3.31) resulta que el cambio de variable tiene la siguiente forma

$$\zeta_x = x_x + M_x \mathcal{C}_{(M_x, S_x)} \Psi_4(d/dt) q_{ox} = x_x - \left( \frac{1}{g} \right) \begin{bmatrix} 0 \\ 0 \\ 1 \\ d/dt \end{bmatrix} q_x. \quad (3.56)$$

---


$${}^4 \mathcal{C}_{(M_x, S_x)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{g} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{g} & 0 & 0 & 0 \end{bmatrix}, \quad \Psi_4(d/dt) = \begin{bmatrix} 1 & 0 & \frac{d}{dt} & 0 & \frac{d^2}{dt^2} & 0 & \frac{d^3}{dt^3} & 0 \\ 0 & 1 & 0 & \frac{d}{dt} & 0 & \frac{d^2}{dt^2} & 0 & \frac{d^3}{dt^3} \end{bmatrix}^T$$


---

La señal de incertidumbre no lineal es

$$q_{x,*} = X_x \mathcal{C}_{(M_x, S_x)} \Psi_4(d/dt) q_{ox} = -\frac{I_{yy}}{g} \left( a_{x,2} + a_{x,1} \frac{d}{dt} + \frac{d^2}{dt^2} \right) q_x + I_{yy} q_\theta. \quad (3.57)$$

De (3.56), (3.39) y (3.55) se obtiene

$$\begin{aligned} \frac{d}{dt} \zeta_x &= A_{F_x} \zeta_x + B_x (\bar{u}_x + q_{x,*}), \\ x &= C_x \zeta_x. \end{aligned} \quad (3.58)$$

Siendo su función de transferencia (recordar (3.41))

$$F_{\zeta_x}(s) = C_x (sI - A_{F_x})^{-1} B_x = -\frac{g}{I_{yy} \pi_{x_x}(s)}. \quad (3.59)$$

Se concluye que con la siguiente ley de control

$$\bar{u}_x = -q_{x,*}, \quad (3.60)$$

el sistema (3.58) se linealiza exactamente.

### Descomposición estructural del subsistema $y - \phi$

De la misma manera que en el caso anterior

- $\ker B_y = 0$ .
- el par  $(A_{F_y}, B_y)$  es controlable.
- el par  $(C_y, A_{F_y})$  es observable.

La solución de la ecuación algebraica  $A_{F_y} M_y + B_y X_y = I_4$  (ver (3.19), (3.25) y (3.20)) es

$$M_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1/g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_y^\ell = \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_{xx} \end{bmatrix}^T, \quad X_y = I_{xx} \begin{bmatrix} a_{y,3}/g \\ a_{y,2}/g \\ a_{y,1} \\ 1 \end{bmatrix}^T. \quad (3.61)$$

### 3.4. Conclusiones

---

Así, de (3.29) y (3.31) el cambio de variable es

$$\zeta_y = x_y + M_y \mathcal{C}_{(M_y, S_y)} \Psi_4(d/dt) q_{oy} = x_y + \begin{pmatrix} 1 \\ -g \end{pmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ d/dt \end{bmatrix} q_y. \quad (3.62)$$

La señal de incertidumbre no lineal tiene la forma

$$q_{y,*} = X_y \mathcal{C}_{(M_y, S_y)} \Psi_4(d/dt) q_{oy} = \frac{I_{xx}}{g} \left( a_{y,2} + a_{y,1} \frac{d}{dt} + \frac{d^2}{dt^2} \right) q_y + I_{xx} q_\phi. \quad (3.63)$$

De (3.62) y (3.43) se obtiene

$$\begin{aligned} \frac{d}{dt} \zeta_y &= A_{F_y} \zeta_y + B_y (\bar{u}_y + q_{y,*}), \\ \mathbf{y} &= C_y \zeta_y. \end{aligned} \quad (3.64)$$

La función de transferencia es (ver (3.45))

$$F_{\zeta_y}(s) = C_y (sI - A_{F_y})^{-1} B_y = \frac{g}{I_{xx} \pi_{x_y}(s)}. \quad (3.65)$$

Concluimos que con la siguiente ley de control

$$\bar{u}_y = -q_{y,*}, \quad (3.66)$$

el sistema (3.64) se linealiza exactamente.

## 3.4 Conclusiones

Se obtuvo el modelo lineal aproximado para cada uno de los subsistemas del cuadricóptero. Esto se logró aplicando la aproximación para ángulos pequeños, lo cual permite simplificar las funciones trigonométricas involucradas. Los productos obtenidos de ángulos, derivadas de ángulos también se desprecian, con esto se consigue que las señales de incertidumbre no lineal sean cero.

Se establecieron las condiciones para construir una ecuación estructural algebraica que permite



obtener un cambio de variable. Este hace posible mapear las señales de incertidumbre no lineal  $q$  de los subsistemas  $x - \theta$  y  $y - \phi$  en nuevas señales de incertidumbre no lineal  $q_*$ .

El cambio de variable no aplica a los subsistemas  $\psi$  y  $z$  porque estos ya tienen la forma (3.30) del Teorema 3.2.1. Antes de realizar el cambio de variable a los subsistemas ya mencionados, se aplicó una realimentación de estado para estabilizar a cada subsistema del cuadricóptero.

# Capítulo 4

## Linealización robusta asintótica por realimentación

En este capítulo se estiman las señales de perturbación no lineal de los subsistemas del cuadricóptero. Para llevar a cabo esta tarea, primero se inicia describiendo el filtro Beard-Jones, el cual es un detector de perturbaciones basado en observadores de estado estándar. Se presenta el análisis en lazo cerrado considerando el subsistema  $x - \theta$ .

En la simulación numérica, se calculan las ganancias de realimentación de estado por medio del diseño LQR. Se presenta el cálculo de las inyecciones de salida de los filtros Beard-Jones, para esto se emplea el método del lugar de las raíces. Los filtros Beard-Jones se llevan a la forma canónica observador. Posteriormente se presentan las gráficas de simulación. Finalmente se aborda la teoría sobre el alargamiento de la vecindad de estabilidad.

### 4.1 Estimación de la no linealidad

Introducimos la siguiente hipótesis.

**H4.** *La descripción en espacio de estados  $\Sigma(A, B, C)$  (3.18) no tiene ceros finitos invariantes*

en el origen. Eso significa:

$$\text{Im } B \cap A \ker C = 0. \quad (4.1)$$

Las suposiciones **H1.3** y **H4** hacen posible aplicar los resultados de [30] y diseñar un rechazo de perturbaciones robusto, basado en el filtro Beard-Jones [31, 32, 33, 34, 35]

$$\begin{aligned} \frac{d}{dt}w(t) &= A_K w(t) - Ky(t) + Bu(t), \\ \hat{q}_*(t) &= -G^\ell(Cw(t) - y(t)), \end{aligned} \quad (4.2)$$

donde  $A_K \doteq (A + KC)$  y (comparar con (3.36))

$$u(t) = -\hat{q}_*(t). \quad (4.3)$$

Aquí  $K : \mathbb{R}^p \rightarrow \mathbb{R}^m$  es la “inyección de salida” tal que (recordar **H1.3**)

$$\sigma\{(A + KC)\} \subset \mathbb{C}^-, \quad (4.4)$$

y  $G^\ell$  expresa la inversa por la izquierda de la ganancia estática  $-C(A+KC)^{-1}B$ . El generador de residuos es

$$\begin{aligned} \frac{d}{dt}e(t) &= A_K e(t) - Bq_*(x, u), \\ \hat{q}_*(t) &= -G^\ell C e(t), \end{aligned} \quad (4.5)$$

donde  $e(t) = w(t) - \zeta(t)$ .

Siendo su función de transferencia

$$F_e(s) = G^\ell C (sI - A_K)^{-1} B. \quad (4.6)$$

Note también que cuando la transformada de Laplace de  $q_*(x, u)$  está bien definida, entonces

$$\hat{q}_*(s) = G^\ell F_e(s) q_*(s). \quad (4.7)$$

### 4.1.1 Sistema en lazo cerrado

A continuación se analiza el sistema en lazo cerrado para el subsistema  $x - \theta$  (el análisis para el subsistema  $y - \phi$  es similar). De (3.58), el filtro (4.2) se reescribe como (recuerde que la salida  $y(t)$  de (3.58) es  $x$ , la entrada  $u(t)$  es  $\bar{u}_x$ , y tomar en cuenta (4.3) y (4.5))

$$\begin{aligned} \frac{d}{dt} w_x &= \left( A_{K_x} + B_x G_x^\ell C_x \right) w_x - \left( K_x + B_x G_x^\ell \right) x, \\ \bar{u}_x &= G_x^\ell C_x w_x - G_x^\ell x, \end{aligned} \quad (4.8)$$

donde

$$A_{K_x} = A_{F_x} + K_x C_x, \quad G_x^\ell = - \left( C_x A_{K_x}^{-1} B_x \right)^\ell,$$

$$K_x = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{x,4} - a_{x_o,4} \\ a_{x,3} - a_{x_o,3} \\ a_{x,2} - a_{x_o,2} \\ a_{x,1} - a_{x_o,1} \end{bmatrix}. \quad (4.9)$$

Con la asignación  $K_x$  se obtiene el polinomio característico de  $A_{K_x}$  (cf. (4.6)) que es un polinomio Hurwitz

$$\pi_{e_x}(s) = \det(s\mathbf{I} - A_{K_x}) = s^4 + a_{x_o,1} s^3 + a_{x_o,2} s^2 + a_{x_o,3} s + a_{x_o,4}. \quad (4.10)$$

La función de transferencia del filtro (4.8) es

$$F_{w_x}(s) = \frac{I_{yy} a_{x_o,4} (s^4 + a_{x,1} s^3 + a_{x,2} s^2 + a_{x,3} s + a_{x,4})}{g s (s^3 + a_{x_o,1} s^2 + a_{x_o,2} s + a_{x_o,3})} = \frac{I_{yy} a_{x_o,4}}{g} \frac{\pi_{x_x}(s)}{s \bar{\pi}_{w_x}(s)}, \quad (4.11)$$

donde

$$\bar{\pi}_{w_x}(s) = s^3 + a_{x_o,1} s^2 + a_{x_o,2} s + a_{x_o,3}, \quad (4.12)$$

es un polinomio Hurwitz.

Considerando  $e_x = w_x - \zeta_x$ , de (4.8) y (3.58) se obtiene la descripción de estado en lazo cerrado

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix} &= A_{CL} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix} + B_{CL} q_{x,*}, \\ x &= C_{CL} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix}. \end{aligned} \quad (4.13)$$

donde

$$A_{CL} = \left[ \begin{array}{c|c} A_{K_x} & 0 \\ \hline B_x G_x^\ell C_x & A_{F_x} \end{array} \right], \quad B_{CL} = \left[ \begin{array}{c} -B_x \\ B_x \end{array} \right], \quad C_{CL} = \left[ 0 \mid C_x \right]. \quad (4.14)$$

Las funciones de transferencia involucradas en la realización en espacio de estados (4.13) son (recordar (4.6) y (3.59))

$$F_{e_x}(s) = G_x^\ell C_x (sI - A_{K_x})^{-1} B_x = \frac{a_{x_o,4}}{\pi_{e_x}(s)} = \frac{a_{x_o,4}}{s^4 + a_{x_o,1}s^3 + a_{x_o,2}s^2 + a_{x_o,3}s + a_{x_o,4}}. \quad (4.15)$$

$$\begin{aligned} F_{CL}(s) &= C_{CL} (sI - A_{CL})^{-1} B_{CL} \\ &= \left[ 0 \mid C_x \right] \left[ \begin{array}{c|c} (sI - A_{K_x})^{-1} & 0 \\ \hline (sI - A_{F_x})^{-1} B_x G_x^\ell C_x (sI - A_{K_x})^{-1} & (sI - A_{F_x})^{-1} \end{array} \right] \left[ \begin{array}{c} -I \\ I \end{array} \right] B_x \end{aligned}$$

$$\begin{aligned} F_{CL}(s) &= F_{\zeta_x}(s) \left( 1 - F_{e_x}(s) \right) = -\frac{g}{I_{yy} \pi_{x_x}(s)} \left( 1 - \frac{a_{x_o,4}}{\pi_{e_x}(s)} \right) = -\frac{g}{I_{yy} \pi_{x_x}(s)} \left( \frac{s \bar{\pi}_{w_x}(s)}{\pi_{e_x}(s)} \right) \\ &= -\frac{g}{I_{yy} \pi_{x_x}(s)} \left( \frac{s(s^3 + a_{x_o,1}s^2 + a_{x_o,2}s + a_{x_o,3})}{s^4 + a_{x_o,1}s^3 + a_{x_o,2}s^2 + a_{x_o,3}s + a_{x_o,4}} \right), \end{aligned} \quad (4.16)$$

donde  $\pi_{x_x}(s)$ ,  $\pi_{e_x}(s)$  y  $\bar{\pi}_{w_x}(s)$  son los polinomios Hurwitz (3.41), (4.10) y (4.12) respectivamente.

Dado que la señal de incertidumbre no lineal  $q_*$  puede tratarse como una señal de perturbación aditiva (ver (3.58)), se pueden utilizar las herramientas lineales para el rechazo de perturbación (como por ejemplo el filtro Beard-Jones (4.8)).

Para sintetizar el filtro Beard-Jones (4.8), la única información que se necesita es la aproximación lineal (3.9), obtenida en la sección 3.1. La función de transferencia (4.16) de la representación en espacio de estados en lazo cerrado (4.13) tiene incorporado el filtro pasa altas  $1 - a_{x_o,4}/\pi_{e_x}(s)$ , por lo que si la señal de incertidumbre no lineal  $q_{x,*}$  tiene un ancho de banda finito, entonces bastará diseñar la frecuencia de corte del filtro pasa altas superior al ancho de banda de  $q_{x,*}$ .

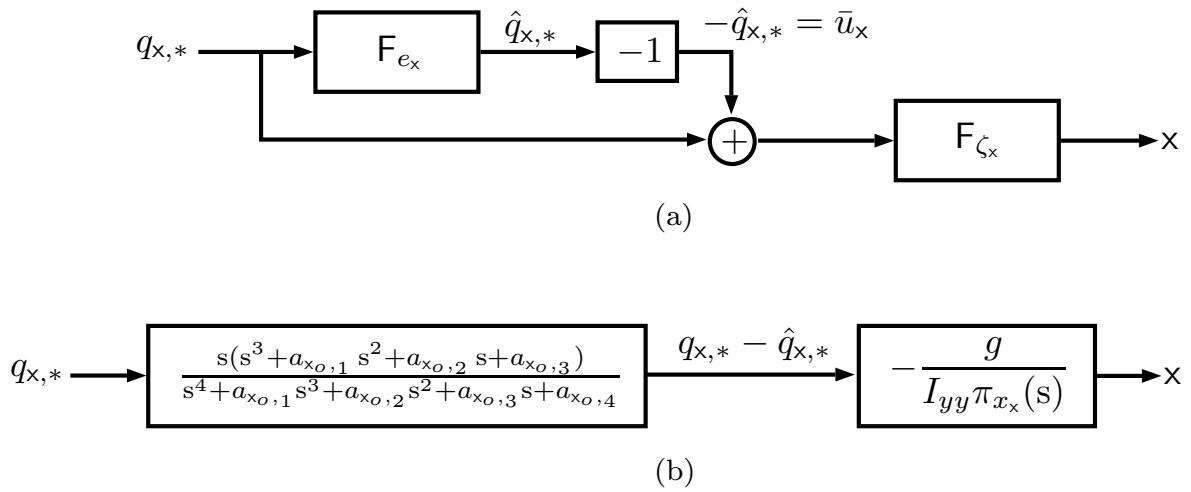


Figura 4.1: Diagrama a bloques de la función de transferencia en lazo cerrado (4.16).

## 4.2 Simulación numérica

Los resultados numéricos se obtienen utilizando Matlab. Considere los siguiente valores obtenidos del cuadricóptero:

	Valor numérico	Unidades
$m$	1.36	kg
$I$	$\begin{bmatrix} 0.0134 & 0 & 0 \\ 0 & 0.0140 & 0 \\ 0 & 0 & 0.0256 \end{bmatrix}$	kg m <sup>2</sup>
$g$	9.81	m s <sup>-2</sup>

Tabla 4.1: Valores numéricos.

El cuadricóptero se llevará a una altura de referencia  $\bar{z} = 2$  m respecto al nivel del suelo

con las siguientes condiciones iniciales:

$$\begin{aligned} \mathbf{x}(0) &= 0 \text{ m}, & \mathbf{y}(0) &= 0 \text{ m}, & \mathbf{z}(0) &= \bar{\mathbf{z}}, \\ d\mathbf{x}(0)/dt &= 0 \text{ m s}^{-1}, & d\mathbf{y}(0)/dt &= 0 \text{ m s}^{-1}, & d\mathbf{z}(0)/dt &= 0 \text{ m s}^{-1}, \end{aligned}$$

$$\begin{aligned} \phi(0) &= \theta(0) = \psi(0) = \alpha_0, \\ d\phi(0)/dt &= d\theta(0)/dt = d\psi(0)/dt = 0 \text{ rad s}^{-1}, \end{aligned}$$

donde  $\alpha_0 = \pi/6$  rad.

### 4.2.1 Realimentaciones de estado

Los vectores de ganancias de realimentación de estado de los subsistemas se obtuvieron por medio del diseño LQR, es decir, la matriz de ganancias de realimentación óptima ( $F_i$  para  $i \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \psi\}$ ) se obtiene resolviendo la ecuación algebraica de Riccati

$$A_i^T P_i + P_i A_i - P_i B_i \rho_i^{-1} B_i^T P_i + Q_i = 0.$$

Para obtener las realimentaciones de estado (3.38), (3.42), (3.46) y (3.50), los valores utilizados de  $Q_i$  y  $\rho_i$  son

$$Q_{\mathbf{x}} = Q_{\mathbf{y}} = 900 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2.25 \end{bmatrix}, \quad \rho_{\mathbf{x}} = \rho_{\mathbf{y}} = 1.$$

$$Q_{\psi} = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}, \quad \rho_{\psi} = 1/12100.$$

$$Q_{\mathbf{z}} = \begin{bmatrix} 1 & 0 \\ 0 & 0.23 \end{bmatrix}, \quad \rho_{\mathbf{z}} = 1/19600.$$

Los vectores de ganancias de realimentación de estado óptimos son

$$F_x = \begin{bmatrix} 30 & 32.43 & -171.92 & -45.05 \end{bmatrix}, \quad (4.17)$$

$$F_y = \begin{bmatrix} -30 & -32.43 & -171.91 & -45.05 \end{bmatrix}, \quad (4.18)$$

$$F_\psi = \begin{bmatrix} -110 & -85.24 \end{bmatrix}, \quad (4.19)$$

$$F_z = \begin{bmatrix} -140 & -69.92 \end{bmatrix}. \quad (4.20)$$

Los espectros de  $A_{F_x}$ ,  $A_{F_y}$ ,  $A_{F_\psi}$  y  $A_{F_z}$  son

$$\Lambda_x(s) = \{-3214.28, -1.91, -0.9535 \pm 1.5852i\},$$

$$\Lambda_y(s) = \{-3358.21, -1.91, -0.9535 \pm 1.5852i\},$$

$$\Lambda_\psi(s) = \{-1.29, -3328.34\},$$

$$\Lambda_z(s) = \{-2.09, -49.32\}.$$

Los radios espectrales de  $\Lambda_x(s)$ ,  $\Lambda_y(s)$ ,  $\Lambda_\psi(s)$  y  $\Lambda_z(s)$  son los siguientes

$$\rho_x = 1.85,$$

$$\rho_y = 1.85,$$

$$\rho_\psi = 1.29,$$

$$\rho_z = 49.32.$$

(4.21)

## 4.2.2 Filtro Beard-Jones

Las inyecciones de salida de los filtros Beard-Jones (4.2) de los subsistemas  $x - \theta$  y  $y - \phi$  se diseñan de tal manera que los polinomios característicos  $\pi_{e_x}(s)$  y  $\bar{\pi}_{w_x}(s)$ , (4.10) y (4.12), sean Hurwitz. Dado que los polinomios  $\bar{\pi}_{w_x}(s)$  y  $\pi_{e_x}(s)$  están relacionados de la siguiente manera:

$$\pi_{e_x}(s) = s \bar{\pi}_{w_x}(s) + a_{x_{o,4}}, \quad (4.22)$$



## Capítulo 4. Linealización robusta asintótica por realimentación

---

entonces se sigue el procedimiento del lugar de las raíces para que ambos polinomios sean Hurwitz.

Mediante este método obtenemos (ver figura 4.2)

$$s(s + 4.75)(s + 4)(s + 3.5) + 28.125 = (s + 1)^2(s^2 + 10.25s + 28.125). \quad (4.23)$$

Es decir los ceros de la función de transferencia,

$$G(s) = 1 - \frac{a_{x_o,4}}{\pi_{e_x}(s)} = \frac{s(s^3 + a_{x_o,1}s^2 + a_{x_o,2}s + a_{x_o,3})}{s^4 + a_{x_o,1}s^3 + a_{x_o,2}s^2 + a_{x_o,3}s + a_{x_o,4}}, \quad (4.24)$$

son

$$\begin{aligned} s_1 &= 0, \\ s_2 &= -4.75, \\ s_3 &= -4, \\ s_4 &= -3.5, \end{aligned} \quad (4.25)$$

y sus polos son

$$\begin{aligned} p_{1,2} &= -1, \\ p_{3,4} &= -5.125 \pm 1.3636j. \end{aligned} \quad (4.26)$$

Note que el par de polos complejos conjugados tienen como frecuencia natural no amortiguada y coeficiente de amortiguamiento

$$\omega_n = 5.3 \text{ rad/s} \quad \text{y} \quad \zeta = 0.966. \quad (4.27)$$

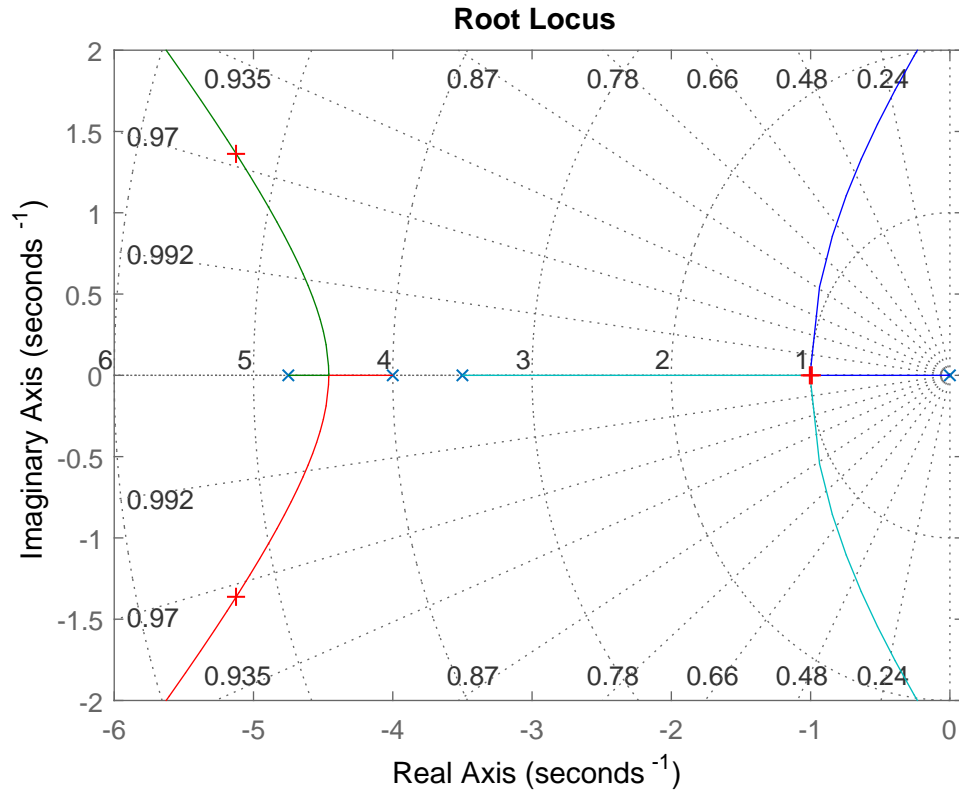


Figura 4.2: Diagrama del lugar geométrico de las raíces.

A fin de facilitar el cálculo de las inyecciones de salida, los filtros Beard-Jones (4.2) se lleva a su forma canónica observador.

### Subsistema $x - \theta$

De (3.58), (4.2) y (4.3) se obtiene la descripción de estado

$$\begin{aligned} \frac{d}{dt} \bar{w}_x &= \left( \bar{A}_{\bar{K}_x} + \bar{B}_x G_x^\ell \bar{C}_x \right) \bar{w}_x - \left( \bar{K}_x + \bar{B}_x G_x^\ell \right) x, \\ \bar{u}_x &= G_x^\ell \bar{C}_x \bar{w}_x - G_x^\ell x, \end{aligned} \tag{4.28}$$

$$\text{donde } \bar{w}_x = T_{ox}^{-1}w_x, \quad T_{ox} = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1}g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1}, \quad \bar{A}_{\bar{K}_x} = \bar{A}_x + \bar{K}_x\bar{C}_x,$$

$$(\bar{A}_x, \bar{B}_x, \bar{C}_x, \bar{K}_x) = (T_{ox}^{-1}A_{F_x}T_{ox}, T_{ox}^{-1}B_x, C_xT_{ox}, T_{ox}^{-1}K_x), \quad G_x^\ell = -\left(C_x(A_{F_x} + K_xC_x)^{-1}B_x\right)^\ell = -\left(\bar{C}_x\bar{A}_{\bar{K}_x}^{-1}\bar{B}_x\right)^\ell \text{ y}$$

$$\bar{A}_x = \begin{bmatrix} 0 & 0 & 0 & -a_{x,4} \\ 1 & 0 & 0 & -a_{x,3} \\ 0 & 1 & 0 & -a_{x,2} \\ 0 & 0 & 1 & -a_{x,1} \end{bmatrix}, \quad \bar{B}_x = \begin{bmatrix} -g/I_{yy} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{C}_x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T, \quad \bar{K}_x = \begin{bmatrix} a_{x,4} - a_{x_o,4} \\ a_{x,3} - a_{x_o,3} \\ a_{x,2} - a_{x_o,2} \\ a_{x,1} - a_{x_o,1} \end{bmatrix},$$

$$G_x^\ell = -\left(\bar{C}_x\bar{A}_{\bar{K}_x}^{-1}\bar{B}_x\right)^\ell = -(a_{x_o,4}I_{yy})/g. \quad (4.29)$$

### Subsistema $y - \phi$

De (3.64), (4.2) y (4.3) obtenemos la descripción de estado

$$\frac{d}{dt}\bar{w}_y = \left(\bar{A}_{\bar{K}_y} + \bar{B}_yG_y^\ell\bar{C}_y\right)\bar{w}_y - \left(\bar{K}_y + \bar{B}_yG_y^\ell\right)y, \quad (4.30)$$

$$\bar{u}_y = G_y^\ell\bar{C}_y\bar{w}_y - G_y^\ell y,$$

$$\text{donde } \bar{w}_y = T_{oy}^{-1}w_y, \quad T_{oy} = \begin{bmatrix} a_{y,3} & a_{y,2} & a_{y,1}g & g \\ a_{y,2} & a_{y,1} & g & 0 \\ a_{y,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1}, \quad \bar{A}_{\bar{K}_y} = \bar{A}_y + \bar{K}_y\bar{C}_y,$$

$$(\bar{A}_y, \bar{B}_y, \bar{C}_y, \bar{K}_y) = (T_{oy}^{-1}A_{F_y}T_{oy}, T_{oy}^{-1}B_y, C_yT_{oy}, T_{oy}^{-1}K_y), \quad G_y^\ell = -\left(C_y(A_{F_y} + K_yC_y)^{-1}B_y\right)^\ell =$$

$$\begin{aligned}
 & - \left( \bar{C}_y \bar{A}_{\bar{K}_y}^{-1} \bar{B}_y \right)^\ell \text{ y} \\
 & \bar{A}_y = \begin{bmatrix} 0 & 0 & 0 & -a_{y,4} \\ 1 & 0 & 0 & -a_{y,3} \\ 0 & 1 & 0 & -a_{y,2} \\ 0 & 0 & 1 & -a_{y,1} \end{bmatrix}, \quad \bar{B}_y = \begin{bmatrix} g/I_{xx} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{C}_y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T, \quad \bar{K}_y = \begin{bmatrix} a_{y,4} - a_{y_{o,4}} \\ a_{y,3} - a_{y_{o,3}} \\ a_{y,2} - a_{y_{o,2}} \\ a_{y,1} - a_{y_{o,1}} \end{bmatrix}, \\
 & G_y^\ell = - \left( \bar{C}_y \bar{A}_{\bar{K}_y}^{-1} \bar{B}_y \right)^\ell = (a_{y_{o,4}} I_{xx})/g.
 \end{aligned} \tag{4.31}$$

### Subsistema $\psi$

De (3.47), (4.2) y (4.3) obtenemos la siguiente descripción de estado

$$\begin{aligned}
 \frac{d}{dt} \bar{w}_\psi &= \left( \bar{A}_{\bar{K}_\psi} + \bar{B}_\psi G_\psi^\ell \bar{C}_\psi \right) \bar{w}_\psi - \left( \bar{K}_\psi + \bar{B}_\psi G_\psi^\ell \right) \psi, \\
 \bar{u}_\psi &= G_\psi^\ell \bar{C}_\psi \bar{w}_\psi - G_\psi^\ell \psi,
 \end{aligned} \tag{4.32}$$

donde  $\bar{w}_\psi = T_{o\psi}^{-1} w_\psi$ ,  $T_{o\psi} = \begin{bmatrix} a_{\psi,1} & 1 \\ 1 & 0 \end{bmatrix}^{-1}$ ,  $\bar{A}_{\bar{K}_\psi} = \bar{A}_\psi + \bar{K}_\psi \bar{C}_\psi$ ,  $(\bar{A}_\psi, \bar{B}_\psi, \bar{C}_\psi, \bar{K}_\psi) = (T_{o\psi}^{-1} A_{F_\psi} T_{o\psi}, T_{o\psi}^{-1} B_\psi, C_\psi T_{o\psi}, T_{o\psi}^{-1} K_\psi)$ ,  $G_\psi^\ell = - \left( C_\psi (A_{F_\psi} + K_\psi C_\psi)^{-1} B_\psi \right)^\ell = - \left( \bar{C}_\psi \bar{A}_{\bar{K}_\psi}^{-1} \bar{B}_\psi \right)^\ell$   
 y

$$\begin{aligned}
 \bar{A}_\psi &= \begin{bmatrix} 0 & -a_{\psi,2} \\ 1 & -a_{\psi,1} \end{bmatrix}, \quad \bar{B}_\psi = \begin{bmatrix} 1/I_{zz} \\ 0 \end{bmatrix}, \quad \bar{C}_\psi = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T, \quad \bar{K}_\psi = \begin{bmatrix} a_{\psi,2} - a_{\psi_{o,2}} \\ a_{\psi,1} - a_{\psi_{o,1}} \end{bmatrix}, \\
 G_\psi^\ell &= - \left( \bar{C}_\psi \bar{A}_{\bar{K}_\psi}^{-1} \bar{B}_\psi \right)^\ell = a_{\psi_{o,2}} I_{zz}.
 \end{aligned} \tag{4.33}$$

### Subsistema $z$

En este subsistema la no linealidad se cancela de manera exacta. Notar que de las expresiones (2.61) y (2.62) se tiene

$$\Delta u_z = -mq_z, \tag{4.34}$$

Sustituyendo (2.45) en (4.34) se llega a

$$\Delta u_z = mg \frac{q_{zz}}{1 + q_{zz}}. \quad (4.35)$$

Considerando el sistema realimentado (3.51), hacemos

$$\bar{u}_z = \Delta u_z. \quad (4.36)$$

para obtener una cancelación exacta de la no linealidad.

Para los subsistemas  $x - \theta$  y  $y - \phi$ , los filtros Beard-Jones se diseñaron tal que sus radios espectrales,  $\varrho_x$  y  $\varrho_y$ , son 10 veces más rápidos que los radios espectrales (4.21), esto es  $\varrho_{e_x} = 10\varrho_x = 18$  y  $\varrho_{e_y} = 10\varrho_y = 18$ . Las ganancias de los filtros son

$$\begin{aligned} \bar{K}_x &= \left[ -2931428.57 \quad -365106.34 \quad -3798.79 \quad 2997.60 \right]^T, \\ \bar{K}_y &= \left[ -2930487.31 \quad -364089.24 \quad -3249.27 \quad 3141.52 \right]^T. \end{aligned}$$

Al escalar los ceros (4.25) y los polos (4.26), se obtiene

$$\begin{aligned} \bar{s}_1 &= 0, \\ \bar{s}_2 &= -85.5, \\ \bar{s}_3 &= -72, \\ \bar{s}_4 &= -63, \end{aligned} \quad (4.37)$$

$$\begin{aligned} \bar{p}_{1,2} &= -18, \\ \bar{p}_{3,4} &= -92.25 \pm 24.5446j. \end{aligned} \quad (4.38)$$

$$\bar{\omega}_n = 95.46 \text{ rad/s} \quad \text{y} \quad \bar{\zeta} = 0.966. \quad (4.39)$$

Para el subsistema  $\psi$ , el filtro Beard-Jones se implementó 40 veces más rápido que el radio espectral  $\varrho_\psi$ , esto es (cf. (4.21))  $\varrho_{e_\psi} = 40\varrho_\psi \approx 52$ . La ganancia del filtro es

$$\bar{K}_\psi = \left[ 1592.87 \quad 3121.63 \right]^T.$$

Para obtener las condiciones iniciales del filtro (4.28) del subsistema  $x - \theta$ , se recurre a la expresión empleada en la obtención de la forma observador del filtro, esto es

$$\bar{w}_x(0) = T_{ox}^{-1} \zeta_x(0), \quad T_{ox}^{-1} = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (4.40)$$

El cambio de variable (3.56) reescrito para (4.40) es

$$\zeta_x(0) = \begin{bmatrix} x(0) \\ dx(0)/dt \\ \theta(0) \\ d\theta(0)/dt \end{bmatrix} - \left(\frac{1}{g}\right) \begin{bmatrix} 0 \\ 0 \\ q_x(0) \\ \frac{d}{dt} q_x(0) \end{bmatrix}. \quad (4.41)$$

De (2.45) y (2.46) se tiene<sup>1</sup>

$$q_x(0) = g\theta(0) - g(\tan \phi(0) \cos \psi(0) + \tan \phi(0) \sec \theta(0) \sin \psi(0)). \quad (4.42)$$

Considerando

$$x(0) = dx(0)/dt = 0,$$

$$\phi(0) = \theta(0) = \psi(0) = \alpha_0,$$

$$d\phi(0)/dt = d\theta(0)/dt = d\psi(0)/dt = 0,$$

de la expresión (4.42) se obtiene

$$\begin{aligned} q_x(0) &= g(\alpha_0 - \tan \alpha_0(\cos \alpha_0 + \tan \alpha_0)), \\ \frac{d}{dt} q_x(0) &= 0. \end{aligned} \quad (4.43)$$

Sustituyendo (4.43) en (4.41) tenemos

$$\zeta_x(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tan \alpha_0(\cos \alpha_0 + \tan \alpha_0). \quad (4.44)$$

---

<sup>1</sup>Note que  $\frac{1}{m} \Delta u_z - g = -\frac{g}{\cos \phi \cos \theta}$

Finalmente, de (4.44) y (4.40) las condiciones iniciales para el filtro son

$$\bar{w}_x(0) = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tan \alpha_0 (\cos \alpha_0 + \tan \alpha_0). \quad (4.45)$$

El procedimiento anterior se aplica de manera similar a (4.30) del subsistema  $y - \phi$

$$\bar{w}_y(0) = T_{oy}^{-1} \zeta_y(0), \text{ esto es, } \bar{w}_y(0) = \begin{bmatrix} a_{y,3} & a_{y,2} & a_{y,1} g & g \\ a_{y,2} & a_{y,1} & g & 0 \\ a_{y,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tan \alpha_0 (1 - \sin \alpha_0).$$

Las condiciones iniciales para el filtro del subsistema  $\psi$  son  $\bar{w}_\psi(0) = T_{o\psi}^{-1} x_\psi(0)$ , esto es,

$$\bar{w}_\psi(0) = \begin{bmatrix} a_{\psi,1} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_0 a_{\psi,1} \\ \alpha_0 \end{bmatrix}.$$

### 4.2.3 Filtro pasa altas del sistema en lazo cerrado

En la sección 4.1.1 se mostró que la función de transferencia en lazo cerrado, entre  $q_{x,*}$  y  $x$  (4.16), tiene incorporado el filtro pasa altas (4.24), que atenúa el efecto de  $q_{x,*}$  en la entrada del sistema en lazo cerrado  $\bar{u}_x$ , ver figura 4.1.

Para los valores obtenidos en la subsección 4.2.2 (*cf.* (4.24) y (4.23)) se tiene

$$\begin{aligned} G(s) &= 1 - \frac{a_{x_o,4}}{\pi_{e_x}(s)} = \frac{s(s^3 + a_{x_o,1}s^2 + a_{x_o,2}s + a_{x_o,3})}{s^4 + a_{x_o,1}s^3 + a_{x_o,2}s^2 + a_{x_o,3}s + a_{x_o,4}} \\ &= \frac{s(s + 85.5)(s + 72)(s + 63)}{(s + 18)^2(s^2 + 184.5s + 9112.5)} \\ &= \frac{s(s^3 + 220.5s^2 + 16078.5s + 387828)}{s^4 + 220.5s^3 + 16078.5s^2 + 387828s + 2952450}. \end{aligned} \quad (4.46)$$

En la figura 4.3 se muestra el diagrama de Bode de (4.46).

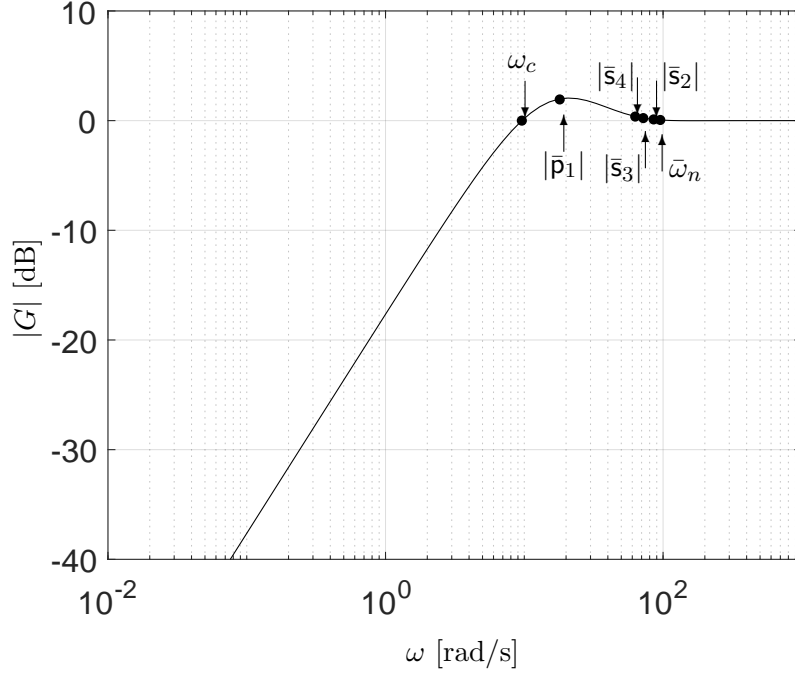


Figura 4.3: Gráfica de Bode de magnitud.  $|\bar{p}_1| = 18$ ,  $|\bar{s}_2| = 85.5$ ,  $|\bar{s}_3| = 72$ ,  $|\bar{s}_4| = 63$ ,  $\bar{\omega}_n = 95.46$  rad/s y  $\bar{\zeta} = 0.966$ .

De (4.46) y de la figura 4.3, se tienen las siguientes observaciones

1. La frecuencia de corte del filtro es

$$\omega_c = \sqrt{\frac{a_{x_o,2}}{2} - \frac{\sqrt{a_{x_o,2}^2 - 2a_{x_o,4}}}{2}} = 9.61 \text{ rad/s.} \quad (4.47)$$

En efecto, haciendo en (4.46)  $|G(j\omega)| = 1$ , se obtiene

$$2a_{x_o,4}\omega^4 - 2a_{x_o,2}a_{x_o,4}\omega^2 + a_{x_o,4}^2 = 0. \quad (4.48)$$

La solución correspondiente a la frecuencia de corte está dada por (4.47) (*cf.* figura 4.3).

2. A altas frecuencias se tiene una ganancia unitaria. En efecto

$$\begin{aligned} \lim_{\omega \rightarrow \infty} |G(j\omega)|^2 &= \lim_{\omega \rightarrow \infty} \frac{(\omega^4 - 16078.5\omega^2)^2 + (-220.5\omega^3 + 387828\omega)^2}{(\omega^4 - 16078.5\omega^2 + 2952450)^2 + (-220.5\omega^3 + 387828\omega)^2}, \\ &= 1. \end{aligned} \quad (4.49)$$



3. Abajo de la frecuencia de corte  $\omega_c$ , se tiene una atenuación de 20 dB por década.
4. La frecuencia  $\omega_{max}$  donde ocurre el máximo<sup>2</sup> de la curva es  $\omega_{max} = 21.021$  rad/s

$$20 \log(|G(j\omega_{max})|) = 20 \log(|G(21.021j)|) = 2.058 \text{ dB} \quad (4.50)$$

5. Note que este pequeño máximo es debido a la sucesión de ceros y polos de (4.46), y no a la presencia de alguna frecuencia de resonancia (*cf.* (4.39) y figura 4.3).
6. Si el ancho de banda de la señal  $q_{x,*}$  (3.57) es inferior a  $\omega_c = 9.61$  rad/s entonces se tendrá una atenuación de 20 dB por década.

#### 4.2.4 Resultados de simulación

En la figura 4.4 se muestran las gráficas obtenidas para el subsistema  $x - \theta$ . De las gráficas 4.4a y 4.4b puede notarse que el sistema se estabiliza en posición ( $x$ ) y en orientación ( $\theta$ ) después de un tiempo de simulación de 5 segundos. La señal de control mostrada en la figura 4.4c al inicio presenta un sobrepaso. En la figura 4.4d se muestra la señal estimada por el filtro Beard-Jones (línea discontinua) y la señal construida (línea continua), puede notarse que el filtro logra estimar la señal que se debe cancelar. Y en la figura 4.4e se muestra la señal de error.

---

<sup>2</sup>Para calcular la frecuencia  $\omega_{max}$  donde ocurre el máximo de la curva, primero se deriva la expresión  $|G(j\omega)|$  con respecto a  $\omega$ , la derivada se iguala a cero y se resuelve para  $\omega$ .

---

## 4.2. Simulación numérica

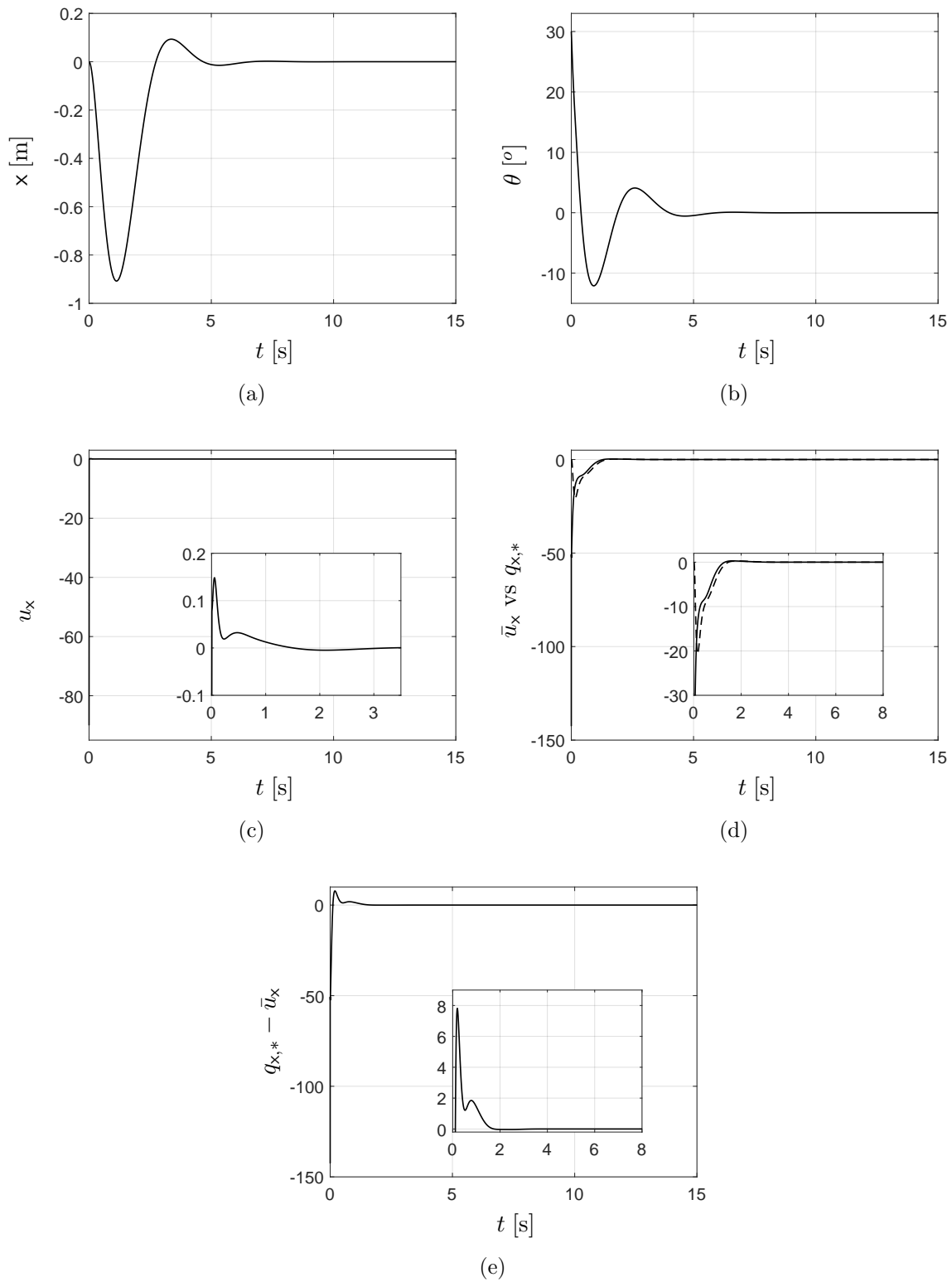


Figura 4.4: Respuestas obtenidas para el subsistema  $x - \theta$ . (a) Posición en  $x$ , (b) posición angular  $\theta$ , (c) entrada de control  $u_x$ , (d)  $\bar{u}_x$  vs  $q_{x,*}$ , (e) error  $q_{x,*} - \bar{u}_x$ .

Las señales obtenidas para el sistema  $y - \phi$  se presentan en la figura 4.5. Note en 4.5a y 4.5a que el subsistema también se estabiliza en posición ( $y$ ) y en orientación ( $\phi$ ) después de un tiempo de simulación de 5 segundos. La señal de control (figura 4.5c) presenta un comportamiento similar al del sistema descrito anteriormente. En la figura 4.5d se observa que el filtro (línea discontinua) estima la señal reconstruida (línea continua) y la figura 4.5e muestra la señal de error.

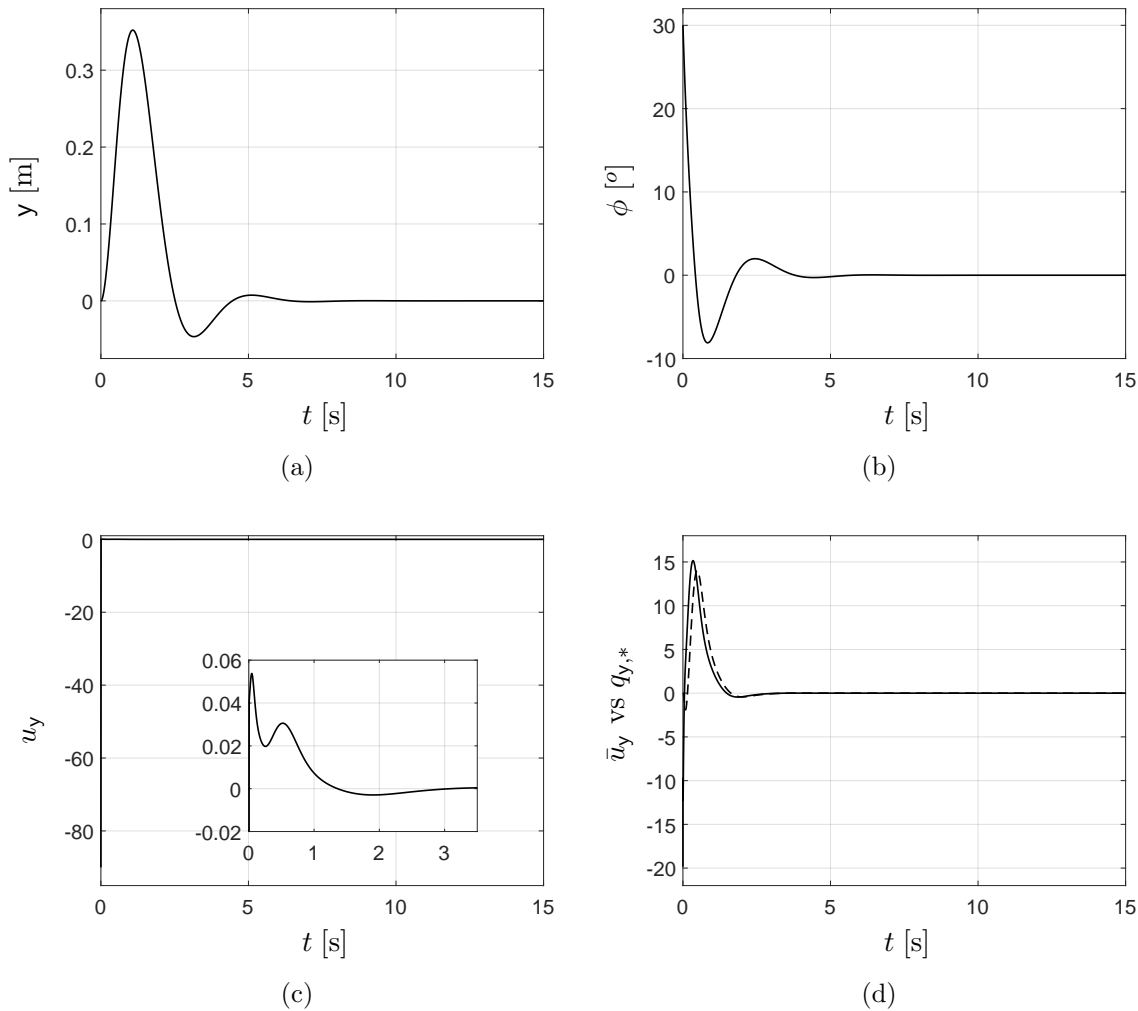


Figura 4.5: Respuestas obtenidas para el subsistema  $y - \phi$ . (a) Posición en  $y$ , (b) posición angular  $\phi$ , (c) entrada de control  $u_y$ , (d)  $\bar{u}_y$  vs  $q_{y,*}$ .

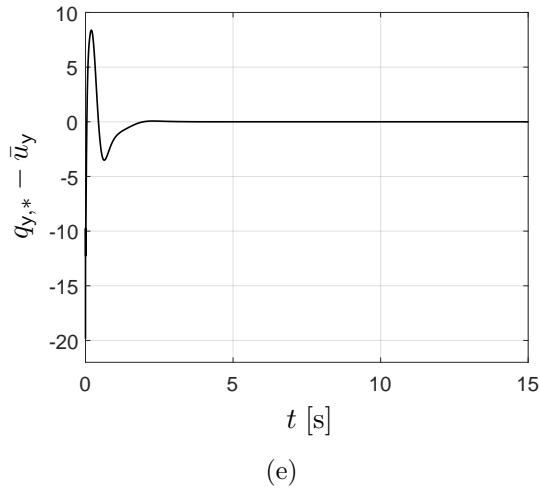


Figura 4.5: Respuestas obtenidas para el subsistema  $y - \phi$ . (e) Error  $q_{y,*} - \bar{u}_y$ .

Las señales obtenidas para el subsistema  $\psi$  se muestran en la figura 4.6. Note que el ángulo de guiñada se estabiliza a los 5 segundos de simulación (figura 4.6a). En la figura 4.6b se presenta la entrada de control. La figura 4.6c muestra la comparación de la señal estimada por el filtro (línea discontinua) y la señal reconstruida (línea continua), la magnitud de esta señal es grande al inicio de la simulación. En la figura 4.6d se muestra la curva de error.

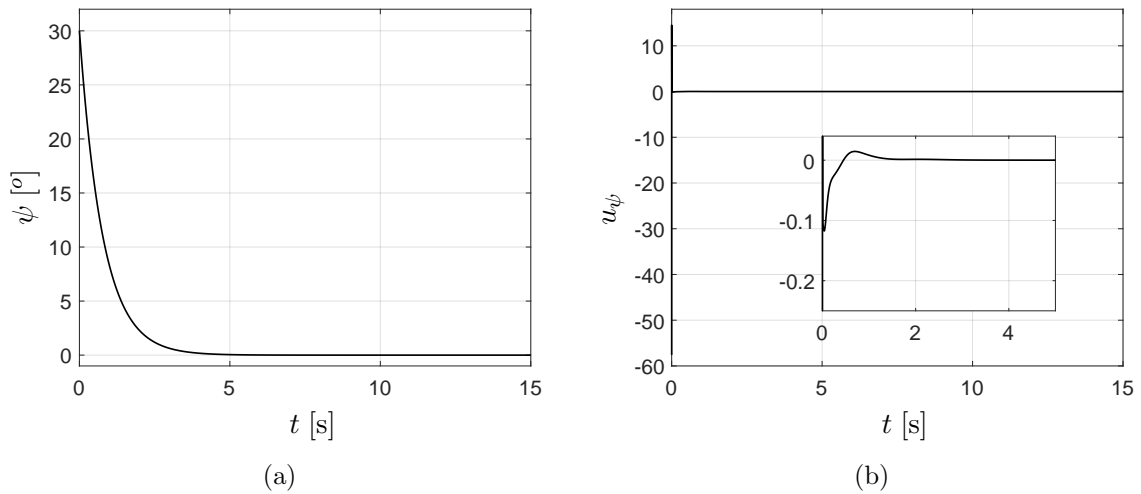


Figura 4.6: Respuestas obtenidas para el subsistema  $\psi$ . (a) Posición angular  $\psi$ , (b) entrada de control  $u_\psi$ .

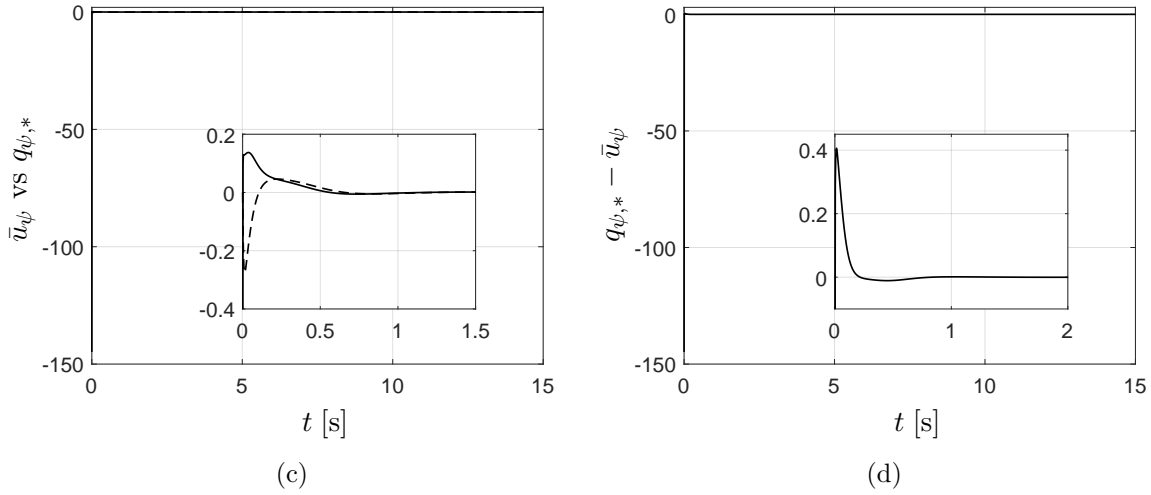


Figura 4.6: Respuestas obtenidas para el subsistema  $\psi$  . (c)  $\bar{u}_{\psi}$  vs  $q_{\psi,*}$ , (d) error  $q_{\psi,*} - \bar{u}_{\psi}$ .

En la figura 4.7 se presentan las graficas de la señal de altura y la entrada de control para el subsistema  $z$ . Note que el cuadricóptero alcanza la señal de referencia de 2 m en un tiempo de simulación de 2.5 s sin presentar sobretiro.

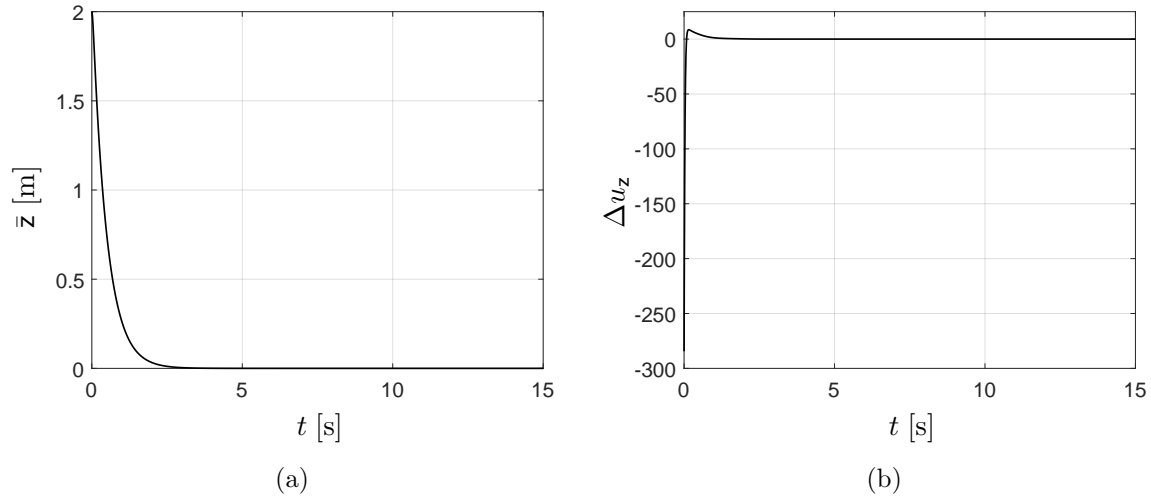


Figura 4.7: Respuestas obtenidas para el subsistema  $z$  . (a) Referencia  $\bar{z} = 2$  m, (b) entrada de control  $\Delta u_z$ .

### 4.2.5 Respuesta en frecuencia

Bajo la suposición de acotación para  $q_*$  y asumiendo que su transformada de Fourier,  $\mathcal{F}\{q_*\}$ , es una función de banda limitada con ancho de banda  $\mathcal{B}_W$ , entonces sintetizamos un filtro Hurwitz pasabajas  $F_e(s)$  con frecuencia de corte  $\omega_c$ , cuyo objetivo es reconstruir las frecuencias bajas de  $q_*$ ; ver figura 4.8. La señal de incertidumbre no lineal  $q_*$  está afectando el comportamiento en lazo cerrado a través del filtro pasa altas  $1 - F_e(s)$ , por lo que la frecuencia de corte  $\omega_c$  del filtro pasabajas  $F_e(s)$  deberá ser lo suficientemente mayor que el ancho de banda  $\mathcal{B}_W$ , para rechazar  $q_*$ , ver figura 4.8 (cf. figura 4.1).

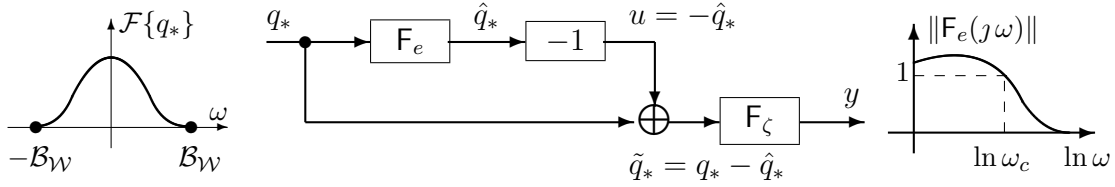


Figura 4.8: Rechazo de la señal de incertidumbre no lineal  $q_*$  con el estimador (4.2) y (4.3).

### 4.2.6 Alargamiento de la vecindad de estabilidad

Asumimos que la máxima razón de cambio (“slew rate”) de la señal de incertidumbre no lineal (3.31) es acotada, es decir

**H5.** Existe  $\mathcal{S}_R \in \mathbb{R}^+$  y  $\|dq_*/dt\|_\infty \leq \mathcal{S}_R$ .

Asumimos que la señal de perturbación no lineal  $q(x, u)$  satisface las siguientes dos propiedades (recordar (3.19), (3.29) y (3.31))

**P1.**  $MC_{(M,S)} \Psi_n(d/dt) q(x, u) = \alpha_M(x)$ .

$\alpha_M(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  es un campo vectorial diferenciable continuo y la matriz  $\partial\alpha_M(x)/\partial x|_{x=0}$

no tiene valor propio  $\lambda = -1$ , i.e.

$$\|\alpha_M(x_1) - \alpha_M(x_2)\| \leq L_M \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{B}_{\rho_1}, \quad L_M > 0, \rho_1 > 0; \quad (4.51)$$

$$\alpha_M(0) = 0; \quad (4.52)$$

$$\det \left( \mathbf{I} + \partial \alpha_M(x) / \partial x \Big|_{x=0} \right) \neq 0. \quad (4.53)$$

**P2.**  $X\mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x, u) = \alpha_X(x) + \Gamma_X(x) u$ .

$\alpha_X(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  y  $\Gamma_X(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$  son campos vectoriales diferenciables continuos y  $\alpha_X(x)$  mapea el vector cero en el vector cero. La constante de Lipschitz de  $\alpha_X(x)$  tiende a cero en la vecindad de origen y  $\Gamma_X(x)$  no tiene valor propio  $\lambda = -1$ , i.e. (ver figura 4.9)

$$\begin{aligned} \|\alpha_X(x_1) - \alpha_X(x_2)\| &\leq L_X \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{B}_\rho, \\ L_X &= \min \left\{ \left( \frac{\mathcal{S}_{\mathcal{R}}}{\rho_0} \right) \rho, \mathcal{S}_{\mathcal{R}} \right\}, \quad \rho \in [0, \rho_1], \end{aligned} \quad (4.54)$$

$$0 < \rho_0 \leq \rho_1;$$

$$\alpha_X(0) = 0; \quad (4.55)$$

$$\det(\mathbf{I} + \Gamma_X(x)) \neq 0 \quad \forall x \in \mathcal{B}_{\rho_1}, \quad \rho_1 > 0. \quad (4.56)$$

donde  $\mathcal{B}_\rho$  designa una vecindad específica de radio  $\rho$ , i.e.

$$\mathcal{B}_\rho = \{ \mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < \rho \}.$$

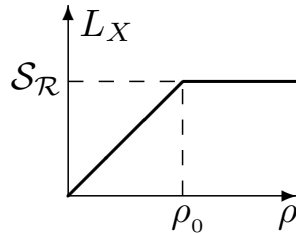


Figura 4.9: Comportamiento de la constante de Lipschitz de  $\alpha_X$ .

## 4.2. Simulación numérica

---

A continuación resumimos tres resultados técnicos que nos permiten concluir sobre el alargamiento de la vecindad de estabilidad<sup>3</sup>.

**Lema 4.2.1.** *Si la hipótesis **H5** es satisfecha, entonces*

$$\|\tilde{q}_*(t)\| \leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + \rho_{SR}, \quad (4.57)$$

donde  $\rho_{SR} = (k_3/\alpha_0) \mathcal{S}_{\mathcal{R}}(\varrho_n^{n-1}/\bar{\omega}_1)$ ,  $\bar{\omega}_1$  es el ancho de banda de (4.2),  $\mathcal{S}_{\mathcal{R}}$  es la razón de cambio de  $q_*$ ,  $\varrho$  es el factor de escala entre el máximo y mínimo valor propio de  $A_K$  y  $k_1$ ,  $k_3$  y  $\alpha_0$  son constantes positivas. Ver figura 4.10.

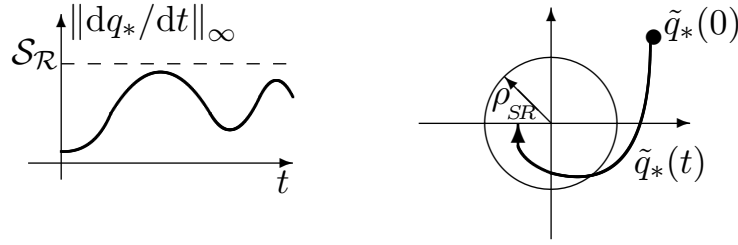


Figura 4.10: Acotamiento de  $\tilde{q}_*$  bajo razón de cambio acotada.

**Teorema 4.2.1.** *Para todo  $x \in \cap_{i=1}^2 \mathcal{B}_{\rho_i}$ , el difeomorfismo*

$$\mathbb{T}_x^\zeta(x) \doteq x + \alpha_M(x), \quad (4.58)$$

satisface

$$\|\mathbb{T}_x^\zeta(x)\| \leq (1 + L_M) \|x\|. \quad (4.59)$$

Para toda  $\zeta \in \mathbb{T}_x^\zeta(\cap_{i=1}^3 \mathcal{B}_{\rho_i})$ , la inversa  $\mathbb{T}_\zeta^x$  satisface

$$(\mathbb{T}_\zeta^x(\mathbb{T}_x^\zeta(x)) = x) : \|\mathbb{T}_\zeta^x(\zeta)\| \leq \mu_0 \|\zeta\|, \mu_0 > 1. \quad (4.60)$$

Ver figura 4.11.

---

<sup>3</sup>Aquí se presenta un resumen de los resultados principales. Ver [36] para los detalles técnicos.



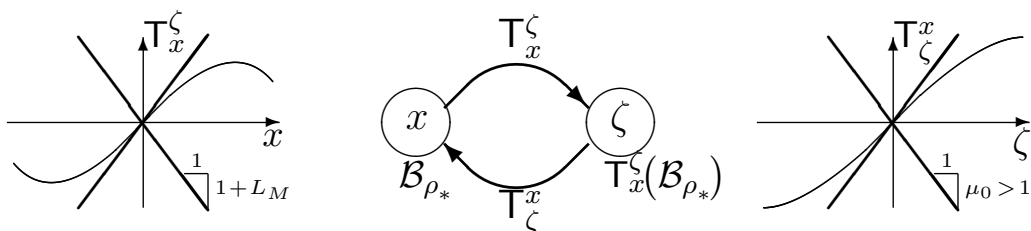


Figura 4.11: Propiedades de los difeomorfismos  $T_x^z$  y  $T_z^x$ .

Si  $A$  es Hurwitz, entonces (3.30) es asintóticamente estable en  $T_x^z(\mathcal{B}_{\rho_*})$ , donde  $\rho_* < \min\{\rho_0, \rho_1, \rho_2, \rho_3, \rho_\#\}$ ,  $\rho_\# = (2\|B\|\bar{\lambda}\mu_0\mathcal{S}_{\mathcal{R}}/\rho_0)^{-1}$ . Ver figura 4.12.

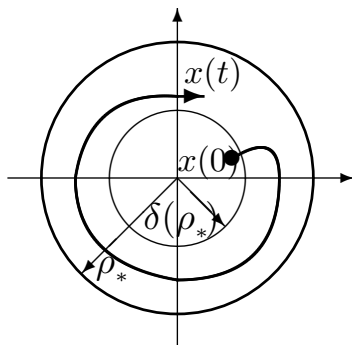


Figura 4.12: Estabilidad de (3.30).

**Lema 4.2.2.** Si la transformada de Fourier de  $q_*$  es de banda limitada, es decir  $\mathcal{F}\{q_*\} = 0$  para toda  $|\omega| > \mathcal{B}_{\mathcal{W}}$ , entonces

$$\|\tilde{q}_*(t)\|_\infty \leq k_1 e^{-(\bar{\omega}_1/2)t} \|\tilde{k}_0\| + \rho_\omega, \quad (4.61)$$

donde  $\rho_\omega = (\mathcal{B}_{\mathcal{W}}/\bar{\omega}_1) k_4 \|q_*(t)\|_\infty$ ;  $\bar{\omega}_1$  es el ancho de banda del filtro Beard-Jones (4.2) y  $\mathcal{B}_{\mathcal{W}}$  es el ancho de banda de la señal de perturbación  $q_*$ . Ver figura 4.13.

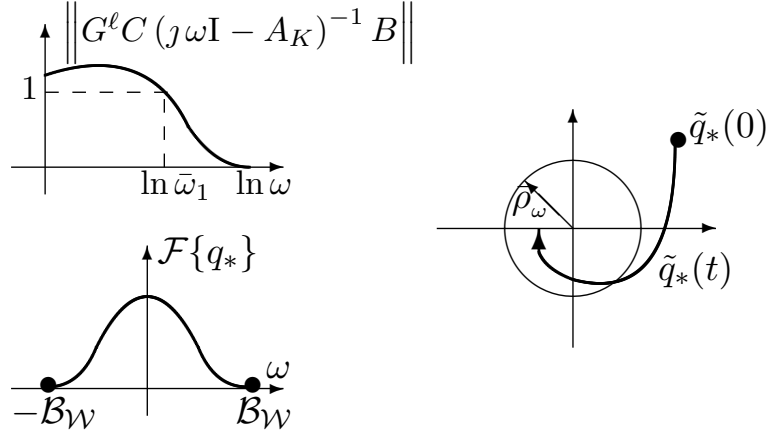


Figura 4.13: Acotamiento de  $\tilde{q}_*$  bajo ancho de banda limitada.

De estos tres resultados se sigue el siguiente corolario:

**Corolario 4.2.1.** *Si el ancho de banda del filtro Beard-Jones (4.2) y el ancho de banda de la señal de perturbación  $q_*$  satisfacen la siguiente desigualdad*

$$\bar{\omega}_1 > \mathcal{B}_W \rho_{\#\#} , \quad (4.62)$$

entonces cualquier trayectoria  $x(t) \in \mathcal{B}_{\bar{\rho}_{**}}$  es atraída hacia la vecindad  $\mathcal{B}_{\underline{\rho}_{**}}$  para  $t > t_{\#\#}$  donde  $\rho_{\#\#} = 4 \|B\| \bar{\lambda} k_{3\#} (\bar{\rho}_{**} / \underline{\rho}_{**})^2 + k_{2\#}$  y  $t_{\#\#} = (2/\omega_c) \log(k_{1\#}(\omega_c/\mathcal{B}_W)/(k_{3\#}\bar{\rho}_{**}))$ . Ver figura 4.14.

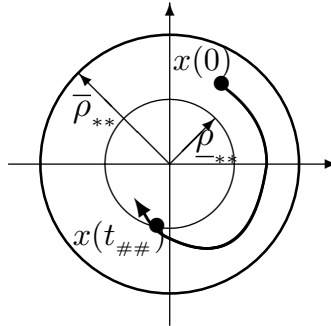


Figura 4.14: Atracción de las trayectorias  $x(t)$  a la vecindad  $\mathcal{B}_{\underline{\rho}_{**}}$ .

Si el ancho de banda del filtro Beard-Jones (4.2) y el ancho de banda de la señal de perturbación  $q_*$  satisfacen la siguiente desigualdad

$$\bar{\omega}_1 > \mathcal{B}_W k_4 \gamma_2, \quad (4.63)$$

entonces

$$\|\tilde{q}_*(t)\|_\infty \leq \bar{k}_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + \bar{\rho}_\omega, \quad (4.64)$$

donde  $\bar{k}_1 = k_1(\bar{\omega}_1/\mathcal{B}_W)/(\bar{\omega}_1/\mathcal{B}_W - k_4 \gamma_2)$  y  $\bar{\rho}_\omega = (k_4 \mu_0 \gamma_1/(\bar{\omega}_1/\mathcal{B}_W - k_4 \gamma_2))L_X \|\zeta(t)\|_\infty$ . Ver figura 4.15.

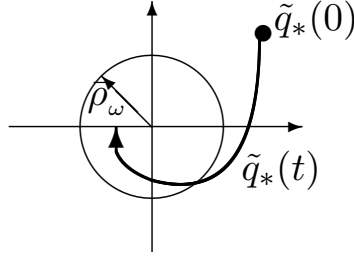


Figura 4.15: Acotamiento de  $\tilde{q}_*$ .

En este Corolario, se muestra que la relación entre la frecuencia de corte  $\omega_c$  del filtro Beard-Jones (4.2) y (4.3),  $\omega_c$ , y el ancho de banda de la señal de incertidumbre no lineal  $q_*$ ,  $\mathcal{B}_W$ , los cuales implican (ver figura 4.16)

- Dado  $0 < \underline{\rho}_{**} < \bar{\rho}_{**}$ , si  $\omega_c > \mathcal{B}_W \rho_{\#\#}$  entonces cualquier trayectoria  $x(t) \in \mathcal{B}_{\bar{\rho}_{**}}$  es atraída hacia la vecindad  $\mathcal{B}_{\underline{\rho}_{**}}$  para  $t > t_{\#\#}$ .
- Si  $\omega_c > \mathcal{B}_W k_{2\#}$  entonces

$$\|q_* - \hat{q}_*\|_\infty \leq \left( \frac{k_{1\#}(\omega_c/\mathcal{B}_W)}{\omega_c/\mathcal{B}_W - k_{2\#}} \right) e^{-\omega_c t/2} + \left( \frac{k_{3\#}}{\omega_c/\mathcal{B}_W - k_{2\#}} \right) \|\zeta\|_\infty, \quad (4.65)$$

### 4.3. Conclusiones

donde  $k_{1\#}$ ,  $k_{2\#}$  y  $k_{3\#}$  son constantes positivas que dependen<sup>4</sup> de las propiedades estructurales de  $q_*$ , y  $\bar{\lambda}$  es el valor propio máximo de la solución de la ecuación de Lyapunov

$$A_F^T P + P A_F = -I$$

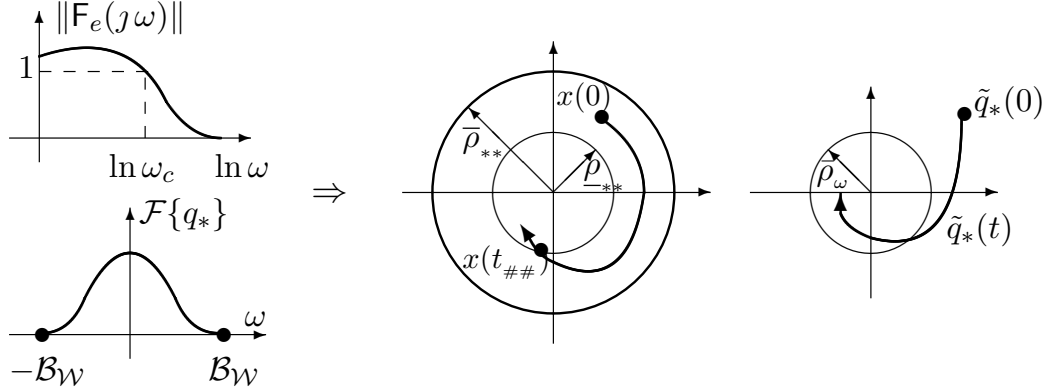


Figura 4.16: Alargamiento de la vecindad de estabilidad  $\tilde{q}_* = q_* - \hat{q}_*$ ,  $\bar{\rho}_\omega = (k_{3\#}/(\omega_c/\mathcal{B}_W - k_{2\#})) \|\zeta\|_\infty$ . Ver corolario 4.2.1.

### 4.3 Conclusiones

Para estimar las señales de perturbación no lineal se utilizó el filtro Beard-Jones, este se describió y se construyó para los subsistemas del cuadricóptero.

En los resultados de simulación numérica se utilizaron valores reales de la matriz de inercia y el peso del cuadricóptero. Las matrices de ganancia de realimentación de estado se obtuvieron al resolver las ecuaciones de Riccati.

La inyección de salida para los subsistemas se calculó mediante el lugar de las raíces. Los filtros Beard-Jones para los subsistemas  $x - \theta$ ,  $y - \phi$  se implementaron 10 veces más rápidos que la dinámica de los subsistemas realimentados. Para el subsistema  $\psi$ , el filtro Beard-Jones se implementó 40 veces más rápido que el sistema realimentado.

<sup>4</sup> $k_{1\#}$  también depende de las condiciones iniciales de  $q_* - \hat{q}_*$ .

Las curvas para los cuatro subsistemas mostraron que el cuadricóptero se estabiliza en orientación mientras alcanza la referencia deseada. Las gráficas también mostraron que los filtros Beard-Jones estimaron las señales de incertidumbre no lineal y con esto se logra una linealización muy cercana a la linealización exacta (dependiendo de la frecuencia de corte del filtro).

Del Principio de Estabilidad de la Primera Aproximación (SPFA, de las siglas en inglés Stability Principle of the First Approximation [37]) sabemos que una realimentación de estado que estabiliza a la linealización (aproximada a los términos de primer orden de su serie de Taylor) de (3.30) (*viz.* (3.18)), es decir, que estabiliza a

$$\begin{aligned}\frac{d}{dt}\zeta(t) &= A\zeta(t) + Bu(t), \\ y(t) &= C\zeta(t),\end{aligned}\tag{4.66}$$

también estabiliza a (3.30) en una vecindad estrecha  $\rho_*$ . Cuando añadimos el estimador de incertidumbre no lineal (4.2) y (4.3), la estabilidad del sistema en lazo cerrado (4.13)

$$\begin{aligned}\frac{d}{dt} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix} &= \begin{bmatrix} A_{K_x} & 0 \\ B_x G_x^\ell C_x & A_{F_x} \end{bmatrix} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix} + \begin{bmatrix} -B_x \\ B_x \end{bmatrix} q_{x,*}, \\ x &= \begin{bmatrix} 0 & C_x \end{bmatrix} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix}.\end{aligned}\tag{4.67}$$

es agrandada.

De hecho, una vecindad atractora deseada  $\mathcal{B}_{\rho_{**}}$  se alcanza por medio de una frecuencia de corte suficientemente grande  $\omega_c$  del filtro Beard-Jones (4.2) y (4.3), ver desigualdad (4.62) del corolario 4.2.1.

Además, con la frecuencia de corte  $\omega_c$  podemos extender la pequeña vecindad inicial  $\rho_*$  a una vecindad extendida más grande  $\rho_{\#\#}$ , y reducir el tiempo transitorio  $t_{\#\#}$ . Es decir, si hacemos  $\omega_c > \mathcal{B}_W \rho_{\#\#}$ , entonces cualquier trayectoria  $x(t)$ , del sistema en lazo cerrado (4.13), dentro de la vecindad  $\mathcal{B}_{\bar{\rho}_{**}}$  es atraída hacia la vecindad de  $\mathcal{B}_{\rho_{**}}$  para  $t > t_{\#\#}$ , ver figura 4.16.

Además, de (4.65) nos damos cuenta de que la norma de error de estimación,  $\|q_* - \hat{q}_*\|_\infty$  puede manejarse arbitrariamente con la relación  $\omega_c/\mathcal{B}_W$ , ver figura 4.16.

---

# Capítulo 5

## Experimentos en exteriores: Vuelo estacionario

En este capítulo se describe el procedimiento para estimar el ancho de banda de la señal de incertidumbre no lineal,  $q_*$ , a partir de mediciones experimentales. Posteriormente se presentan los resultados experimentales de vuelo estacionario del cuadricóptero. Estos resultados se obtienen al aplicar las realimentaciones de estado más los estimadores de señal de incertidumbre no lineal. Se hace una comparación con las señales obtenidas al emplear únicamente realimentaciones de estado estabilizantes. Las pruebas de vuelo estacionario para obtener ambos resultados se llevaron a cabo en exteriores.

### 5.1 Realimentación localmente estabilizante

Para estabilizar localmente (2.55), (2.57), (2.59) y (2.61), se usa la técnica de control LQR. Para esto, se resuelve la ecuación algebraica de Riccati (EAR)

$$A_i^T P_i + P_i A_i - P_i B_i (\rho_i \mathbf{I})^{-1} B_i^T P_i + Q_i = 0, \quad i \in \{x, y, z, \psi\}. \quad (5.1)$$

Para la selección de las matrices de ponderación  $Q_i$ , se analizaron diversas sintonizaciones exitosas de controladores proporcional-derivativo. Analizando estas sintonizaciones exitosas,

se buscó una ponderación de la matriz  $Q_i$ , que condujera a realimentaciones de estado semejantes a tales sintonizaciones exitosas. Se encontró que la siguiente opción de  $Q_i$  proporciona un buen comportamiento del cuadricóptero

$$Q_x = Q_y = 900 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2.25 \end{bmatrix}, \quad Q_z = \begin{bmatrix} 1 & 0 \\ 0 & 0.23 \end{bmatrix}, \quad Q_\psi = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}. \quad (5.2)$$

El ajuste de  $\rho_i$  se realizó mediante un procedimiento de prueba y error en experimentos de campo, obteniéndose de esta manera

$$(\rho_x, \rho_y, \rho_z, \rho_\psi) = (1, 1, 1/19600, 1/12100). \quad (5.3)$$

Resolviendo la EAR (5.1) para las matrices (2.56), (2.58), (2.60) y (2.62), con la elección de (5.2) y (5.3), se obtienen las realimentaciones localmente estabilizantes ( $i \in \{x, y, z, \psi\}$ )

$$u_i = F_i x_i + \bar{u}_i, \quad i \in \{x, y\}, \quad (5.4)$$

$$u_i = F_i(x_i - \bar{x}_i), \quad i \in \{z, \psi\}, \quad \bar{x}_z = \begin{bmatrix} \bar{z} \\ 0 \end{bmatrix}, \quad \bar{x}_\psi = 0. \quad (5.5)$$

$$F_x = \begin{bmatrix} 30 & 32.43 & -171.92 & -45.05 \end{bmatrix}, \quad F_z = \begin{bmatrix} -140 & -69.92 \end{bmatrix}, \quad (5.6)$$

$$F_y = \begin{bmatrix} -30 & -32.43 & -171.91 & -45.05 \end{bmatrix}, \quad F_\psi = \begin{bmatrix} -110 & -85.24 \end{bmatrix}.$$

Con las realimentaciones de estado (5.4), (5.5) y (5.6), se obtienen los siguientes espectros y polinomios característicos

$$\Lambda_x(s) = \{-3214.28, -1.91, -0.9535 \pm 1.5852i\}, \quad \Lambda_z(s) = \{-2.09, -49.32\}, \quad (5.7)$$

$$\Lambda_y(s) = \{-3358.21, -1.91, -0.9535 \pm 1.5852i\}, \quad \Lambda_\psi(s) = \{-1.29, -3328.34\}.$$

$$\pi_{x_i}(s) = \det(sI - A_{F_i}) = s^4 + a_{i,1}s^3 + a_{i,2}s^2 + a_{i,3}s + a_{i,4}, \quad i \in \{x, y\}, \quad (5.8)$$

$$\pi_{x_i}(s) = \det(sI - A_{F_i}) = s^2 + a_{i,1}s + a_{i,2}, \quad i \in \{z, \psi\},$$

donde  $A_{F_i} = A_i + B_i F_i$ ,  $i \in \{x, y, z, \psi\}$ . Note que la dinámica dominante de  $\Lambda_x$  y  $\Lambda_y$  está acotada por  $\omega_\ell = \sqrt{0.9535^2 + 1.5852^2} = 1.85$  rad/s.

## 5.2 Estimador de señal de incertidumbre no lineal

Dado que el cuadricóptero está planeado para moverse en el plano  $(x, y)$  a una altura fija  $z = \bar{z}$ , se sintetiza el observador de señal de incertidumbre no lineal (4.2) y (4.3) para (2.55) y las matrices (2.56), esto es, los filtros Beard-Jones utilizados en la simulación numérica (ver sección 4.2): (4.28) y (4.29), para el subsistema  $x - \theta$ , y (4.30) y (4.31) para el subsistema  $y - \phi$ .

En la sección 4.3, se mencionó que con la frecuencia de corte de los filtros Beard-Jones,  $\omega_c$ , se puede modular tanto la vecindad de linealidad como el error de estimación. Es decir, si la frecuencia de corte  $\omega_c$  es suficientemente mayor al ancho de banda  $\mathcal{B}_W$  de la señal de incertidumbre no lineal  $q_*$ , entonces las vecindades de linealidad y los errores de estimación alcanzarán los valores deseados.

A continuación se muestra un procedimiento de sintonización de los filtros Beard-Jones basado en el comportamiento frecuencial del cuadricóptero. Para esto se procede de la siguiente manera:

1. Se muestran las relaciones existentes entre los polinomios característicos implicados en los filtros Beard-Jones.
2. Se analiza la respuesta en frecuencia del filtro pasa altas incorporado en las funciones de transferencia entre las señales de incertidumbre no lineal  $q_{i,*}$ ,  $i \in \{x, y\}$  y sus salidas,  $x$  e  $y$ .
3. Se estima el espectro en frecuencia de las señales de incertidumbre no lineal  $q_{i,*}$ ,  $i \in \{x, y\}$ .



### 5.2.1 Relación entre los polinomios característicos

Los polinomios característicos de  $A_{K_i}$ ,  $i \in \{x, y\}$ , y las funciones de transferencia en lazo cerrado son (cf. (4.10), (4.11) y (4.12)) además de (4.15)

$$\pi_{e_i}(s) = s^4 + a_{i_o,1}s^3 + a_{i_o,2}s^2 + a_{i_o,3}s + a_{i_o,4}, \quad (5.9)$$

$$\pi_{w_i}(s) = s\bar{\pi}_{w_i}(s) = s(s^3 + a_{i_o,1}s^2 + a_{i_o,2}s + a_{i_o,3}), \quad (5.10)$$

$$F_{e_i}(s) = \frac{a_{i_o,4}}{\pi_{e_i}(s)}, \quad (5.11)$$

donde  $A_{K_i} = A_{F_i} + K_i C_i$ ,  $\pi_{e_i}(s) = \det(sI_4 - A_{K_i})$ ,  $\pi_{w_i}(s) = \det(sI_4 - (A_{K_i} + B_i G_i^\ell C_i))$ ,  $F_{e_i}(s) = G_i^\ell C_i (sI - A_{K_i})^{-1} B_i$ ,  $i \in \{x, y\}$  (cf. sección 4.1.1).

De (5.9) y (5.10), obtenemos (cf. (4.22))

$$\pi_{e_i}(s) = s\bar{\pi}_{w_i}(s) + a_{i_o,4}, \quad (5.12)$$

Cuando se siguió el procedimiento del lugar de las raíces, se obtuvo (cf. (4.23))

$$s\bar{\pi}_{w_i}(s) = s(s + 4.75\varrho_c)(s + 4\varrho_c)(s + 3.5\varrho_c), \quad (5.13)$$

$$\pi_{e_i}(s) = (s + \varrho_c)^2(s^2 + 10.25\varrho_c s + 28.125\varrho_c^2). \quad (5.14)$$

Note que

$$\begin{aligned} a_{i_o,1} &= 12.25\varrho_c, \\ a_{i_o,2} &= 49.625\varrho_c^2, \\ a_{i_o,3} &= 66.50\varrho_c^3, \\ a_{i_o,4} &= 28.125\varrho_c^4. \end{aligned} \quad (5.15)$$

De esta manera se obtienen dos polinomios Hurwitz,  $\pi_{e_i}(s)$  y  $\bar{\pi}_{w_i}(s)$ , parametrizados con la constante positiva  $\varrho_c$ .

## 5.2.2 Respuesta en frecuencia

En las secciones 4.1.1 y 4.2.5 se mostró que la función de transferencia en lazo cerrado (4.16) tiene incorporado un filtro pasa altas (recordar figuras 4.1 y 4.8).

Cuando se parametriza a los filtros de la sección 4.2.3, con la constante positiva  $\varrho_c$ , se obtiene (cf. (4.46) y recordar (5.11) y (5.12))

$$G(s) = 1 - F_{e_i}(s) = 1 - \frac{a_{i_o,4}}{\pi_{e_i}(s)} = \frac{s \bar{\pi}_{w_i}(s)}{\pi_{e_i}(s)}, \quad (5.16)$$

$$G(j\omega) = \frac{(\omega^2 - a_{i_o,2})\omega^2 - j(a_{i_o,1}\omega^2 - a_{i_o,3})\omega}{((\omega^2 - a_{i_o,2})\omega^2 + a_{i_o,4}) - j(a_{i_o,1}\omega^2 - a_{i_o,3})\omega}. \quad (5.17)$$

La frecuencia de corte  $\omega_c$  del filtro pasa altas  $G(j\omega)$  es (cf. (4.47) y recordar (5.14))

$$\omega_c = \sqrt{\frac{a_{i_o,2}}{2} - \frac{\sqrt{a_{i_o,2}^2 - 2a_{i_o,4}}}{2}} = 0.5339\varrho_c \text{ rad/s}. \quad (5.18)$$

El parámetro  $\varrho_c$  debe ajustarse suficientemente más grande que el ancho de banda  $\mathcal{B}_{\mathcal{Y}}$  de  $\mathcal{F}\{q_{i,*}\}$ ,  $i \in \{x, y\}$ , ver figura 5.1 y compare con figura 4.3.

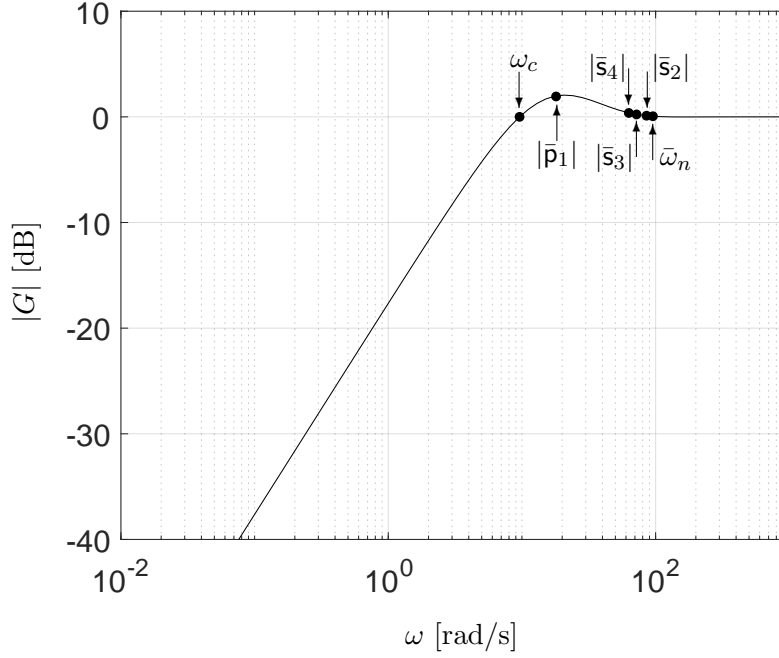


Figura 5.1: Gráfica de Bode de  $|G(j\omega)|_{dB}$ , (5.17) y (5.14) con  $\varrho_c = 18$ .  $|\bar{p}_1| = 18$ ,  $|\bar{s}_2| = 85.5$ ,  $|\bar{s}_3| = 72$ ,  $|\bar{s}_4| = 63$ ,  $\bar{\omega}_n = 95.46 \text{ rad/s}$ ,  $\bar{\zeta} = 0.966$  y  $\omega_c = 9.61 \text{ rad/s}$ ;  $|G(0)| = 0$  y  $|G(\infty)| = 1$ .

### 5.2.3 Estimación del ancho de banda

Para estimar el ancho de banda de  $q_{i,*}$ ,  $i \in \{x, y\}$ , se tomaron datos en tiempo real de las posiciones  $x$  e  $y$  mientras el cuadricóptero se mantiene en vuelo estacionario únicamente con las realimentaciones de estado (4.17)–(4.20). En la figura 5.2a se muestra la evolución en el tiempo de la variable  $y$  en vuelo estacionario. En la figura 5.2b se muestra la transformada de Fourier de  $y$ , obtenida con la herramienta FFT de MATLAB<sup>®</sup>. En la figura 5.2b podemos notar que el ancho de banda de  $y$  es 0.4 Hz, esto es, 2.51 rad/s. Haciendo  $\varrho_c = 18$ , de (5.18) se obtienen las frecuencias de corte de los filtros Beard-Jones  $\omega_c = 9.61 \text{ rad/s}$  (cf. (4.47)). Entonces los polinomios  $\pi_{e_i}(s)$  toman los valores (cf. (5.14))

$$\begin{aligned} \pi_{e_i}(s) &= (s + 18)^2(s^2 + 184.5s + 9112.5), \\ &= s^4 + 220.5s^3 + 16078.5s^2 + 387828s + 2952450. \end{aligned} \quad (5.19)$$

### 5.3. Vuelo estacionario en exteriores

---

De esta forma la frecuencia de corte del filtro Beard-Jones,  $\omega_c$ , es 3.83 veces más grande que el ancho de banda  $\mathcal{B}_y$  de la transformada de Fourier de  $y$ . De (5.19) y (5.15) se puede apreciar que el coeficiente  $a_{i_o,4}$  crece en un factor  $\varrho_c^4$ , por lo que en la práctica  $\varrho_c$  no puede ser muy grande. El valor  $\varrho_c = 18$ , mostró buenos resultados experimentales en exteriores.

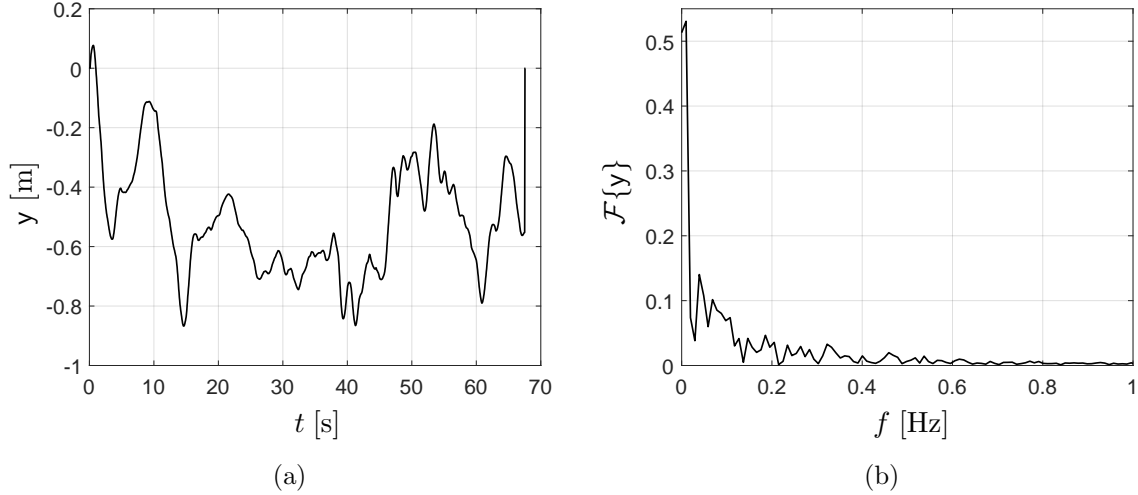


Figura 5.2: (a) Posición en  $y$  obtenida en vuelo estacionario y usando realimentación de estado, (b) espectro de amplitud en el dominio de la frecuencia de  $y$ .

### 5.3 Vuelo estacionario en exteriores

Para probar la eficiencia del estimador de la señal de incertidumbre no lineal (4.28)–(4.31), se realizaron experimentos de vuelo estacionario en exteriores. El marco de referencia fijo al cuerpo se encuentra a una altura  $z = 4.3$  m sobre el nivel del suelo. En vuelo estacionario se desea permanecer en  $(x, y, z) = (0, 0, 4.3)$  m y  $(\phi, \theta, \psi) = (0, 0, 0)$  [°].

A continuación se presentan los resultados experimentales y las comparaciones en  $x$ ,  $y$  y  $\psi$ , esto es

1. Cuando se aplican las realimentaciones estabilizantes  $F_x x_x$ ,  $F_y x_y$  y  $F_\psi x_\psi$  más la estimación de la señal de incertidumbre no lineal  $\bar{u}_x$ ,  $\bar{u}_y$  y  $\bar{u}_\psi$ .

2. Cuando únicamente se aplican las realimentaciones estabilizantes  $F_x x_x$ ,  $F_y x_y$  y  $F_\psi x_\psi$ .

### 5.3.1 Subsistema $x - \theta$

En las figuras 5.3a y 5.3b se presenta el comportamiento de la señal de posición  $x$  y la señal posición angular  $\theta$ , respectivamente. De las figuras 5.3a y 5.3b se puede observar que el ángulo trata de compensar las variaciones de  $x$ , debidas a la presencia de la perturbación del aire. En la figura 5.3c se muestra la señal de realimentación de estado. En la figura 5.3d se presenta el comportamiento de la señal de incertidumbre no lineal obtenida mediante el filtro Beard-Jones.

### 5.3. Vuelo estacionario en exteriores

---

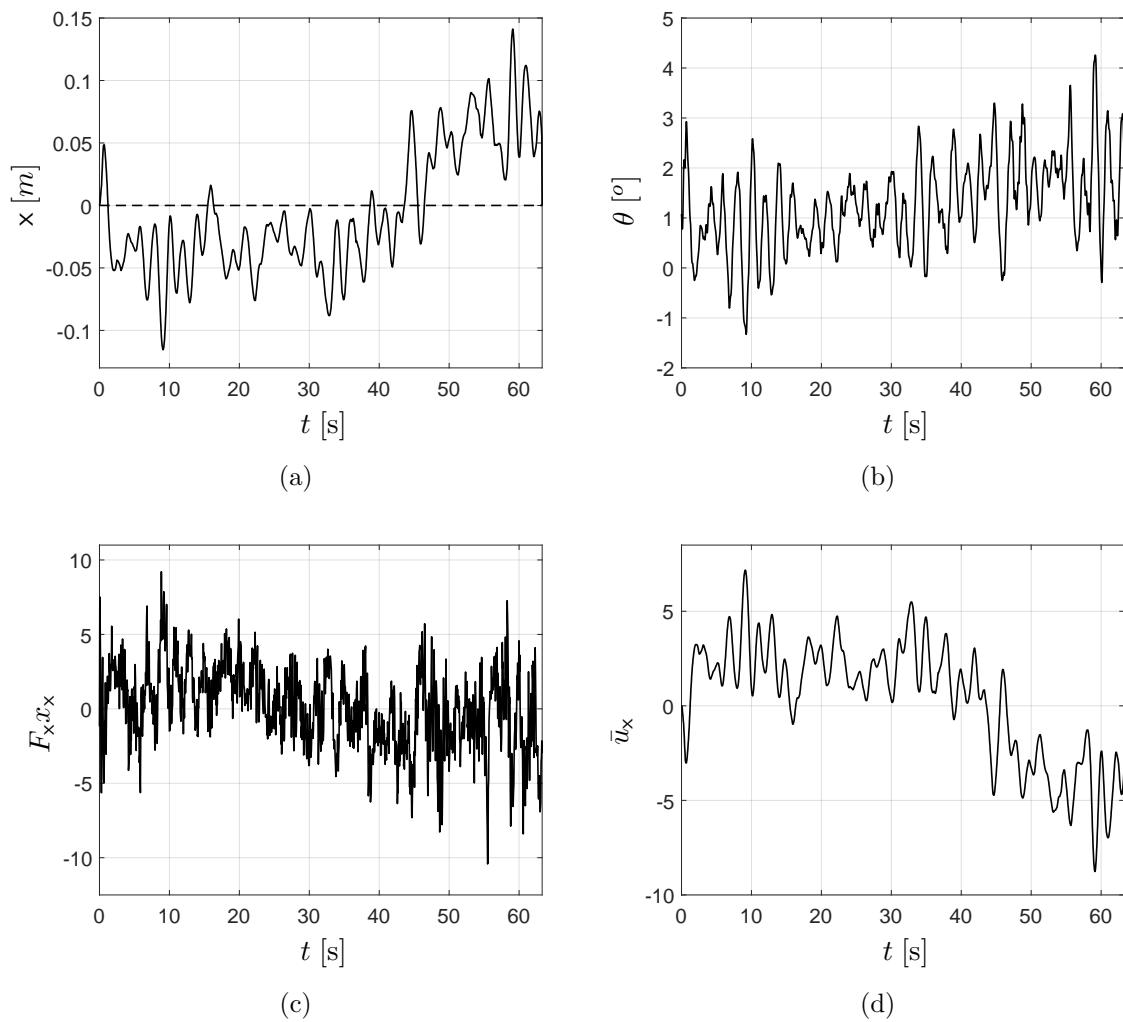


Figura 5.3: Subsistema  $x - \theta$ . (a) Posición en  $x$ , (b) ángulo de cabeceo  $\theta$ , (c) señal de realimentación estabilizante  $F_x x_x$ , (d) señal estimada  $\bar{u}_x$ .

En la figura 5.4 se comparan las señales de posición en  $x$ . La gráfica representada mediante la línea discontinua se obtiene al aplicar únicamente la realimentación de estado (5.4) (con  $\bar{u}_x = 0$ ). La gráfica en línea continua se obtiene al aplicar la realimentación de estado junto con el estimador de señal de incertidumbre no lineal (5.4) y (4.28) con (4.29). En esta figura puede observarse el efecto compensador del control propuesto.

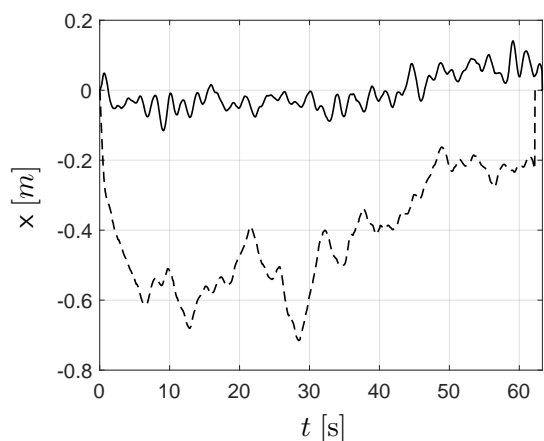


Figura 5.4: Comparación de las señales de posición  $x$  aplicando realimentación de estado (línea discontinua) y realimentación más el filtro Beard-Jones (línea continua) para el subsistema  $x - \theta$ .

### 5.3.2 Subsistema $y - \phi$

El comportamiento de la señal de posición  $y$  y la señal de posición angular  $\phi$  se muestra en las figuras 5.5a y 5.5b respectivamente. En estas gráficas podemos notar que la posición  $y$  se mantiene sobre la referencia mientras que la señal del ángulo de alabeo se encuentra cerca de la referencia.

En la figura 5.5c se presenta la señal de realimentación de estado. El comportamiento de la señal estimada mediante el filtro Beard-Jones se muestra en la figura 5.5d.

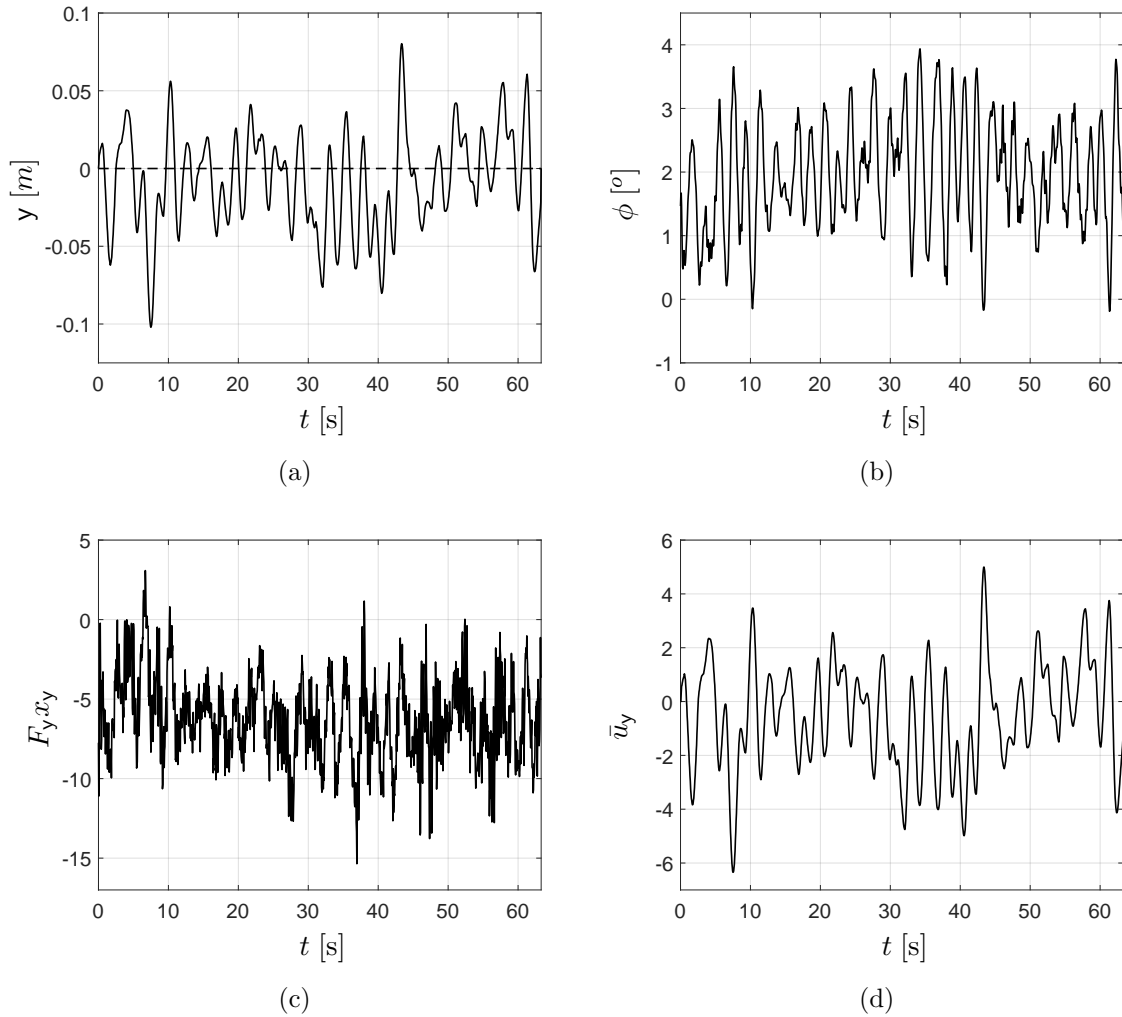


Figura 5.5: Subsistema  $y - \phi$ . (a) Posición en  $y$ , (b) ángulo de alabeo  $\phi$ , (c) señal de realimentación estabilizante  $F_y x_y$ , (d) señal estimada  $\bar{u}_y$ .

En la figura 5.6 se presenta la comparación de las señales de posición en  $y$ . La gráfica representada por la línea discontinua se obtiene al aplicar la realimentación de estado (5.4) (con  $\bar{u}_y = 0$ ). La gráfica en línea continua se obtiene al aplicar la realimentación de estado más el estimador de señal de incertidumbre no lineal (5.4) y (4.30) con (4.31).



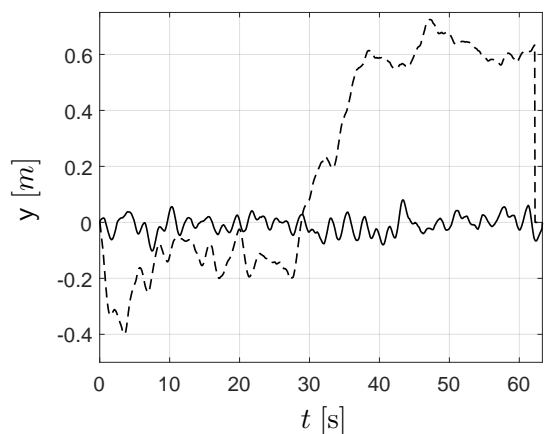


Figura 5.6: Comparación de las señales de posición  $y$  y aplicando realimentación de estado (línea discontinua) y realimentación más el filtro Beard-Jones (línea continua) para el subsistema  $y - \phi$ .

### 5.3.3 Subsistema $\psi$

En la figura 5.7a se muestra la comparación de las señales del ángulo de guiñada  $\psi$ . La gráfica en línea discontinua se obtiene al aplicar únicamente la realimentación de estado. La gráfica en línea continua se obtiene al aplicar la realimentación de estado más la señal obtenida con el filtro Beard-Jones. En la figura 5.7b se muestra la señal de realimentación de estado. La señal obtenida con el filtro Beard-Jones se presenta en la figura 5.7c.

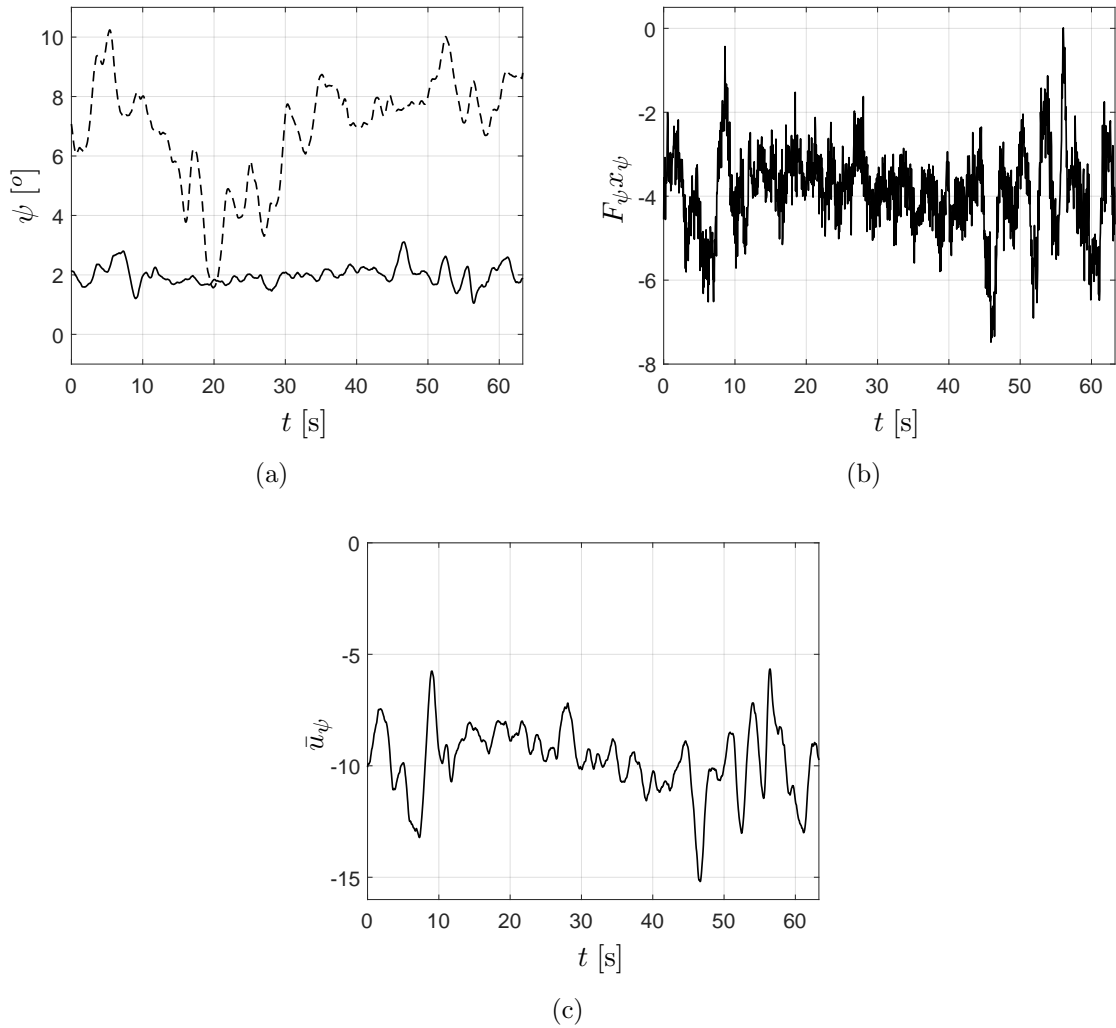


Figura 5.7: Subsistema  $\psi$ . (a) Comparación de las señales del ángulo de guiñada aplicando realimentación de estado (línea discontinua) y realimentación más el filtro Beard-Jones (línea continua), (b) señal de realimentación estabilizante  $F_{\psi} x_{\psi}$ , (c) señal estimada  $\bar{u}_{\psi}$ .

### 5.3.4 Subsistema $z$

A este subsistema solo se aplicó una realimentación de estado. En la figura 5.8a se muestra el comportamiento de la señal de altura para una referencia de 4.3 m. Es importante mencionar que para las pruebas de vuelo estacionario, el cuadricóptero se vuela de forma manual y la

altura el usuario la establece.

En la figura 5.8b se presenta la señal de realimentación de estado.

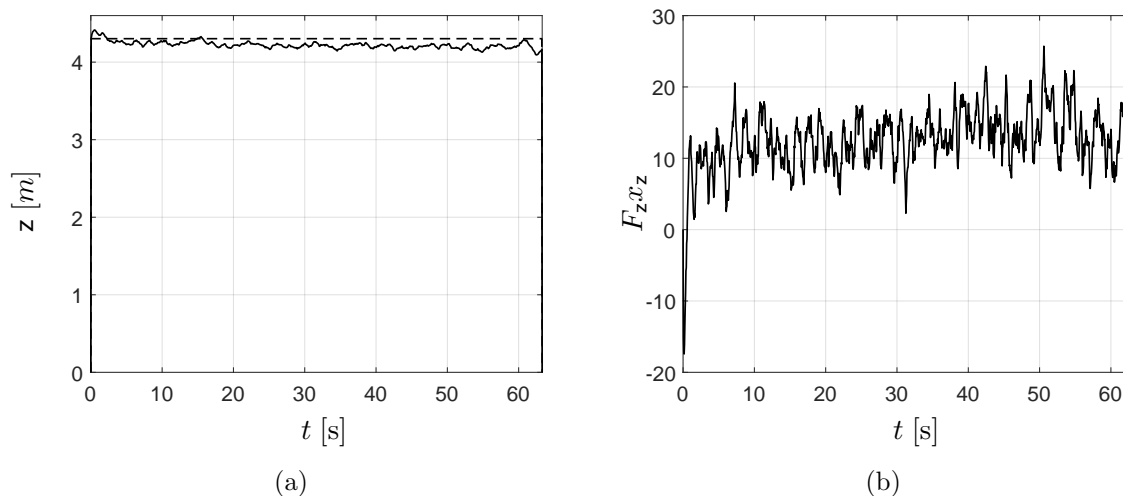


Figura 5.8: Subsistema  $z$ . (a) Altura  $z = 4.3$  m, (b) señal  $F_z x_z$ .

## 5.4 Conclusiones

Se describió el procedimiento para obtener un estimado del ancho de banda de la señal de incertidumbre no lineal, esto es, la sintonización de los filtros Beard-Jones. Para esto se examinó la relación entre los polinomios característicos involucrados en los filtros Beard-Jones y se introdujo la constante de parametrización  $\rho_c$ . Se analizó la respuesta en frecuencia del filtro pasa altas y posteriormente se estimó el ancho de banda de la señal  $q_{i,*}$ ,  $i \in \{x, y\}$ , para esto se emplearon datos experimentales de la señal de posición en  $y$ , en vuelo estacionario. Con la ayuda de MATLAB<sup>®</sup> se obtuvo el espectro de amplitud en el dominio de la frecuencia. Aplicando las realimentaciones de estado más los estimadores de señal de incertidumbre no lineal, se obtienen los resultados experimentales mostrados en las figuras 5.3, 5.5 y 5.7 para los subsistemas  $x - \theta$ ,  $y - \phi$  y  $\psi$  respectivamente. Puede notarse que las señales de posición y orientación se mantienen cerca de la referencia deseada. En las figuras 5.4, 5.6 y 5.7a las señales  $x$ ,  $y$  y  $\psi$  se comparan con las señales obtenidas al aplicar únicamente las

#### 5.4. Conclusiones

---

realimentaciones de estado. Puede notarse que el esquema de control propuesto, LQR más el filtro Beard-Jones, tiene un mejor desempeño que solamente aplicar el LQR; esto se debe al efecto linealizante del filtro Beard-Jones.



# Capítulo 6

## Seguimiento de trayectoria

En este capítulo se presenta un esquema de seguimiento robusto para su implementación en exteriores en un cuadricóptero. Se inicia con la caracterización del ruido del viento, el cual causa el fenómeno de deriva. Para eliminar el fenómeno de deriva, el polo en el origen del estimador de incertidumbre se desplaza ligeramente a la izquierda del plano complejo, obteniéndose así un estimador sin deriva. Se presenta el correspondiente análisis en lazo cerrado y su respuesta en frecuencia. Se describe la síntesis de la trayectoria de estado óptima, la trayectoria se particiona en un conjunto finito de puntos estacionarios locales. Finalmente se presentan los resultados experimentales realizados en exteriores.

### 6.1 Ruido del viento

Como se ha visto en los resultados experimentales del capítulo anterior, los estimadores de señal de incertidumbre no lineal (4.28), (4.30) y (4.32) tiene un buen desempeño cuando el cuadricóptero está en vuelo estacionario. Sin embargo, cuando deseamos que el cuadricóptero siga alguna trayectoria en el plano  $(x, y)$ , a veces el cuadricóptero sufre un fenómeno de deriva. Esto se debe principalmente al ruido del viento.

Para caracterizar el ruido del viento, hemos procedido como sigue

1. Primero se guardan los datos de las señales  $x$  e  $y$  del vuelo estacionario en exteriores, controlado mediante la aplicación de las realimentaciones estabilizantes locales (5.4), (5.5) y (5.6), y el estimador de incertidumbre no lineal (4.28)–(4.31) y (5.19) (ver figura 6.1a).
2. Se aplica la transformada de Fourier (ver figura 6.1c).
3. Basado en la forma de la señal posición  $y$  (ver figura 6.1a), se propone la siguiente señal de ruido

$$n(t) = a_1 \sin(\omega_1 t) + a_2 \sin(b_1 \pi (\sin(\omega_2 t) + b_2)) + a_0. \quad (6.1)$$

4. Se ajustan los coeficientes de la señal de ruido del viento propuesta en (6.1), para obtener los picos principales de la señal y por medio de la transformada rápida de Fourier (ver figura 6.1c). Las posiciones principales de los picos se obtiene ajustando  $b_1$ ,  $b_2$ ,  $\omega_1$  y  $\omega_2$ ; y las magnitudes principales de los picos se obtienen ajustando  $a_0$ ,  $a_1$  y  $a_2$ . Entonces obtenemos

$$n(t) = 0.009 \sin(0.85t) + 0.0115 \sin(2.1\pi(\sin(0.4t) + 1)) + 0.0073; \quad (6.2)$$

ver figura 6.1b, y compare la figura 6.1d con la figura 6.1c.

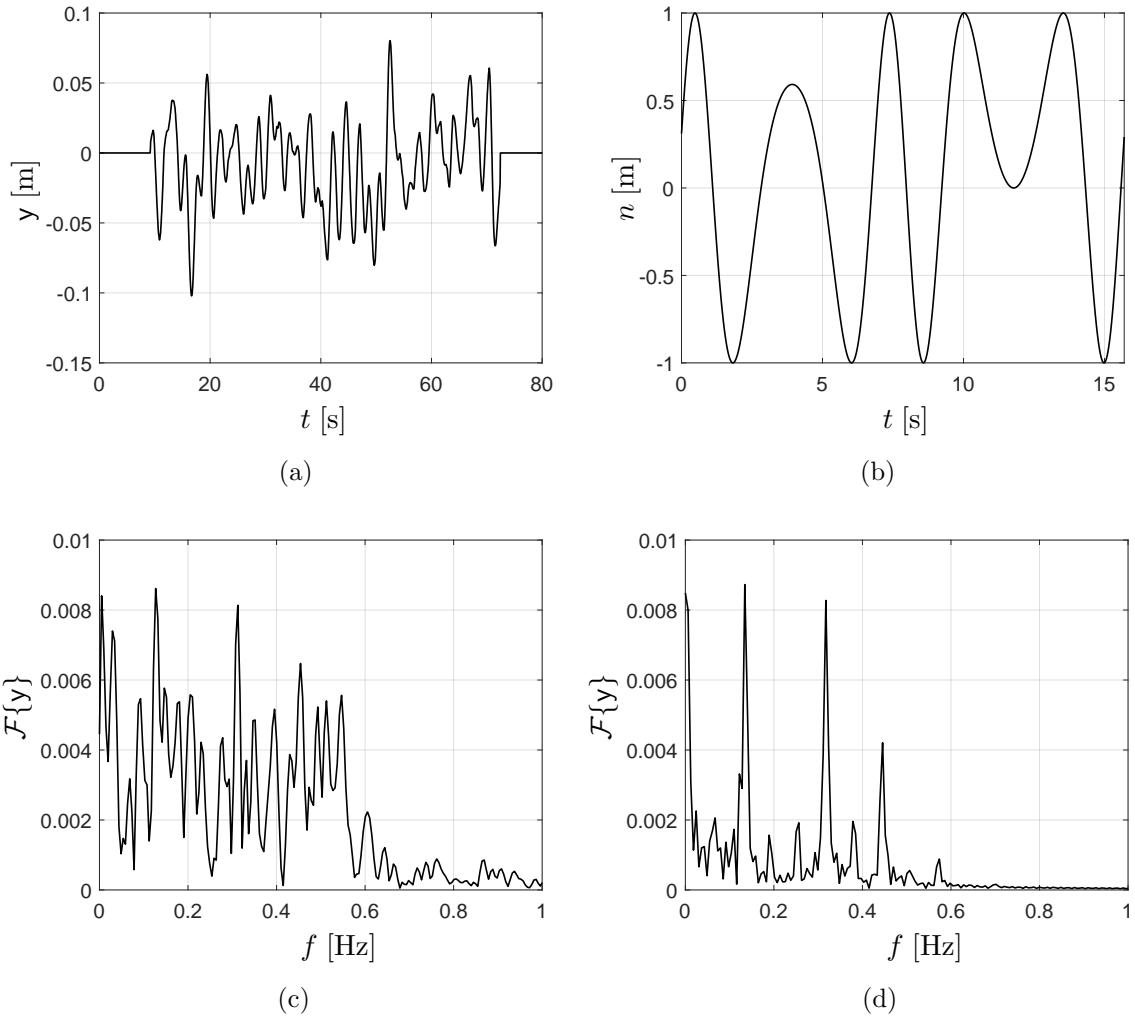


Figura 6.1: Caracterización del ruido del viento. (a) Posición en  $y$  [m], (b) estimación de ruido del viento  $n$  [m], (c) transformada rápida de Fourier de  $y$ , (d) transformada rápida de Fourier de  $n$ .

Analizando la estimación del ruido del viento (6.2), deducimos

- (i) El término constante 0.0073 es fácilmente compensado por el término integral del estimador de incertidumbre no lineal, ver (5.10).
- (ii) El término sinusoidal  $0.009 \sin(0.85t)$  es atenuado por el filtro pasa altas  $1 - F_e(s)$ , ver (4.16) y recordar figura 5.1.



Entonces, el término  $\sin(2.1\pi(\sin(0.4t) + 1))$  es la causa del fenómeno de deriva.

## 6.2 Fenómeno de deriva

Para esto analizamos el término

$$\eta_d(t) = \sin\left(2.1\pi\left(\sin\left(\frac{2\pi t}{T}\right) + 1\right)\right). \quad (6.3)$$

Cuando el término (6.3) pasa a través de la acción integral del estimador de incertidumbre no lineal (4.28)–(4.31) (recordar (5.10)), obtenemos para  $(\mu - 1)T < t \leq \mu T$

$$I_t = \int_0^t \eta_d(\tau) d\tau = \sum_{\kappa=1}^{\nu-1} \int_{(\kappa-1)T}^{\kappa T} \eta_d(\tau) d\tau + \int_{(\nu-1)T}^t \eta_d(\tau) d\tau = (\nu - 1)I_T + R_t, \quad (6.4)$$

donde  $-T \leq R_t \leq T$ . Por lo tanto, si  $I_T > 0$  entonces  $\lim_{\nu \rightarrow \infty} I_t = \infty$ .

Integrando (6.3) con  $T = 15.7$  s, podemos constatar que  $I_T > 0$  (ver figura 6.2a), y por consiguiente  $\lim_{\nu \rightarrow \infty} I_t = \infty$  (ver figura 6.2b).

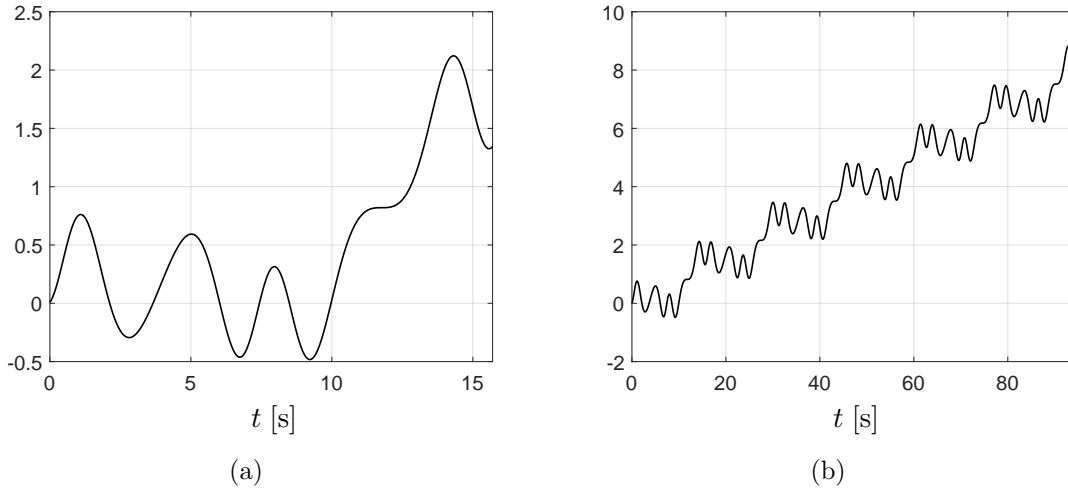


Figura 6.2: Fenómeno de deriva. (a) , (b)  $\int_0^t \sin(2.1\pi(\sin(2\pi\tau/T)+1))d\tau$ ,  $T = \pi/0.2 = 15.7$  s; (a)  $0 \leq t \leq T$ . (b)  $0 \leq t \leq 6T$ .

### 6.3 Estimación sin deriva

Para superar la deriva del cuadricóptero debido a la integración del término (6.3) del ruido del viento modelado por (6.2), hemos procedido como en [38, 39]; es decir, hemos desplazado el polo del origen de (4.8), *viz* (4.28), ligeramente a la izquierda del plano complejo. Esto es, la raíz en el origen del polinomio característico (5.10) es desplazada a la izquierda. Para esto se le agrega el parámetro  $\epsilon a_{i_o,4}$ , donde  $\epsilon$  es una constante positiva suficientemente pequeña. Entonces (5.10) se puede aproximar de la manera siguiente

$$\begin{aligned}
s(s^3 + a_{i_o,1}s^2 + a_{i_o,2}s + a_{i_o,3}) + \epsilon a_{i_o,4} &= s^4 + a_{i_o,1}s^3 + a_{i_o,2}s^2 + a_{i_o,3}s + \epsilon a_{i_o,4} \\
&\approx s^4 + \left(a_{i_o,1} + \epsilon \frac{a_{i_o,4}}{a_{i_o,3}}\right)s^3 + \left(a_{i_o,2} + \epsilon \frac{a_{i_o,4}a_{i_o,1}}{a_{i_o,3}}\right)s^2 \\
&\quad + \left(a_{i_o,3} + \epsilon \frac{a_{i_o,4}a_{i_o,2}}{a_{i_o,3}}\right)s + \epsilon a_{i_o,4} \\
&= \left(s + \epsilon \frac{a_{i_o,4}}{a_{i_o,3}}\right)(s^3 + a_{i_o,1}s^2 + a_{i_o,2}s + a_{i_o,3}).
\end{aligned} \tag{6.5}$$

Esta modificación del polinomio característico (5.10) se logra al reducir ligeramente la ganancia estática  $G^\ell$  del estimador de señal de incertidumbre no lineal (4.8), esto es

$$\begin{aligned}
\frac{d}{dt}w_{df} &= (A_K + (1 - \epsilon)BG^\ell C)w_{df} - (K + (1 - \epsilon)BG^\ell)y, \\
\bar{u}_{df} &= (1 - \epsilon)G^\ell(Cw_{df} - y),
\end{aligned} \tag{6.6}$$

Note que el polinomio característico de  $(A_K + (1 - \epsilon)BG^\ell C)$  no tiene raíces en el origen, en efecto

$$\begin{aligned}
\pi_{w_{df}}(s) &= \det(sI_n - (A_K + (1 - \epsilon)BG^\ell C)) \\
\pi_{w_{df}}(0) &= \det(-A_K - (1 - \epsilon)BG^\ell C) \\
&= \det A_K \det(-I_n - (1 - \epsilon)A_K^{-1}BG^\ell C) \\
\frac{\pi_{w_{df}}(0)}{\det A_K} &= \det(-I_p - (1 - \epsilon)CA_K^{-1}BG^\ell) \\
&= \det(-I_p + (1 - \epsilon)I_p) \\
&= \det(-\epsilon I_p) = (-\epsilon)^p \neq 0.
\end{aligned} \tag{6.7}$$

### 6.3.1 Estimador sin deriva para el subsistema $x - \theta$ : Forma observador

La representación en la forma canónica observador del filtro Beard-Jones (6.6) para el subsistema  $x - \theta$  es (cf. (4.28))

$$\begin{aligned} \frac{d}{dt} \bar{w}_{df,x} &= \left( \bar{A}_{\bar{K}_x} + (1 - \epsilon) \bar{B}_x G_x^\ell \bar{C}_x \right) \bar{w}_{df,x} - \left( \bar{K}_x + (1 - \epsilon) \bar{B}_x G_x^\ell \right) x, \\ \bar{u}_{df,x} &= (1 - \epsilon) G_x^\ell (\bar{C}_x \bar{w}_{df,x} - x), \end{aligned} \quad (6.8)$$

donde las matrices  $\bar{A}_x$ ,  $\bar{B}_x$ ,  $\bar{C}_x$ ,  $\bar{K}_x$  y  $G_x^\ell$  están dadas en (4.29).

### 6.3.2 Estimador sin deriva para el subsistema $y - \phi$ : Forma observador

La representación en la forma canónica observador del filtro Beard-Jones (6.6) para el subsistema  $y - \phi$  es (cf. (4.30))

$$\begin{aligned} \frac{d}{dt} \bar{w}_{df,y} &= \left( \bar{A}_{\bar{K}_y} + (1 - \epsilon) \bar{B}_y G_y^\ell \bar{C}_y \right) \bar{w}_{df,y} - \left( \bar{K}_y + (1 - \epsilon) \bar{B}_y G_y^\ell \right) y, \\ \bar{u}_{df,y} &= (1 - \epsilon) G_y^\ell (\bar{C}_y \bar{w}_{df,y} - y), \end{aligned} \quad (6.9)$$

donde las matrices  $\bar{A}_y$ ,  $\bar{B}_y$ ,  $\bar{C}_y$ ,  $\bar{K}_y$  y  $G_y^\ell$  están dadas en (4.31).

### 6.3.3 Estimador sin deriva para el subsistema $\psi$ : Forma observador

La representación en la forma canónica observador del filtro Beard-Jones para el subsistema  $\psi$  es (cf. (4.32))

$$\begin{aligned} \frac{d}{dt} \bar{w}_{df,\psi} &= \left( \bar{A}_{\bar{K}_\psi} + (1 - \epsilon) \bar{B}_\psi G_\psi^\ell \bar{C}_\psi \right) \bar{w}_{df,\psi} - \left( \bar{K}_\psi + (1 - \epsilon) \bar{B}_\psi G_\psi^\ell \right) \psi, \\ \bar{u}_{df,\psi} &= (1 - \epsilon) G_\psi^\ell (\bar{C}_\psi \bar{w}_{df,\psi} - \psi), \end{aligned} \quad (6.10)$$

donde las matrices  $\bar{A}_\psi$ ,  $\bar{B}_\psi$ ,  $\bar{C}_\psi$ ,  $\bar{K}_\psi$  y  $G_\psi^\ell$  están dadas en (4.33).

## 6.4 Sistema en lazo cerrado

De manera similar que en la subsección 4.1.1, a continuación se presenta el análisis del sistema en lazo cerrado para el subsistema  $x - \theta$  (el análisis para el subsistema  $y - \phi$  es análogo). De (6.6) y (3.58), el sistema en lazo cerrado está representado por (cf. (4.13))

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} e_{df,x} \\ \zeta_x \end{bmatrix} &= A_{CL_{df,x}} \begin{bmatrix} e_{df,x} \\ \zeta_x \end{bmatrix} + B_{CL} q_{x,*}, \\ x &= C_{CL} \begin{bmatrix} e_{df,x} \\ \zeta_x \end{bmatrix}, \end{aligned} \quad (6.11)$$

$$A_{CL_{df,x}} = \left[ \begin{array}{c|c} A_{K_x} & 0 \\ \hline (1 - \epsilon) B_x G_x^\ell C_x & A_{F_x} \end{array} \right], \quad (6.12)$$

donde  $e_{df,x} = w_{df,x} - \zeta_x$  y las matrices  $B_{CL}$  y  $C_{CL}$  están dadas como en (4.14).

### 6.4.1 Función de transferencia

La función de transferencia de  $\Sigma(A_{CL_{df,x}}, B_{CL}, C_{CL})$  es (cf. (4.16))

$$F_{CL_{df,x}}(s) = C_{CL} \left( sI - A_{CL_{df,x}} \right)^{-1} B_{CL} = F_{\zeta_x}(s) (1 - F_{e_{df,x}}(s)), \quad (6.13)$$

donde

$$F_{e_{df,x}}(s) = (1 - \epsilon) G_x^\ell C_x (sI - A_{K_x})^{-1} B_x. \quad (6.14)$$

Comparando (6.14) con (4.6), se puede notar que a la función de transferencia del generador de residuos (4.5) se le ha agregado el término  $(1 - \epsilon)$ .

De (6.6) y (6.14) obtenemos (recordar (6.7) y comparar con (5.10) y (5.11))

$$\pi_{w_{df,x}}(s) = \det(sI_4 - ((A_{F_x} + K_x C_x) + (1 - \epsilon) B_x G_x^\ell C_x)) \quad (6.15)$$

$$\pi_{w_{df,x}}(s) = s \bar{\pi}_{w_x}(s) + \epsilon a_{x_o,4}, \quad (6.16)$$

$$F_{e_{df,x}}(s) = \frac{(1 - \epsilon) a_{x_o,4}}{\pi_{e_x}(s)}, \quad (6.17)$$

### 6.4.2 Respuesta en frecuencia

La función de transferencia en lazo cerrado (6.13) tiene incorporado un filtro pasa altas (recordar (6.17), (5.12) y (6.16))

$$\mathbf{G}_{\text{HPF}_{df,x}}(s) = 1 - \mathbf{F}_{e_{df,x}}(s) = 1 - \frac{(1 - \epsilon)a_{x_{o,4}}}{\pi_{e_x}(s)} = \frac{s \bar{\pi}_{w_x}(s) + \epsilon a_{x_{o,4}}}{\pi_{e_x}(s)}, \quad (6.18)$$

$$\mathbf{G}_{\text{HPF}_{df,x}}(j\omega) = \frac{((\omega^2 - a_{x_{o,2}})\omega^2 + \epsilon a_{x_{o,4}}) - j(a_{x_{o,1}}\omega^2 - a_{x_{o,3}})\omega}{((\omega^2 - a_{x_{o,2}})\omega^2 + a_{x_{o,4}}) - j(a_{x_{o,1}}\omega^2 - a_{x_{o,3}})\omega}. \quad (6.19)$$

La frecuencia de corte  $\omega_{c_{df,x}}$  del filtro pasa altas  $\mathbf{G}_{\text{HPF}_{df,x}}(j\omega)$  para  $\epsilon = 1/50$  es (recordar (5.14))<sup>1</sup>

$$\omega_{c_{df}} = \sqrt{\frac{a_{i_{o,2}}}{2} - \frac{\sqrt{a_{i_{o,2}}^2 - 2(1 + \epsilon)a_{i_{o,4}}}}{2}} = 0.5392\varrho_c \text{ rad/s}. \quad (6.20)$$

En la figura 6.3 se muestra la gráfica de Bode de  $\mathbf{G}_{\text{HPF}_{df,x}}(j\omega)$ , (6.19), con  $\varrho_c = 18$  y  $\epsilon = 1/50$  (recordar (5.15)). Notemos que el comportamiento de las altas frecuencias es muy similar al mostrado en la figura 5.1 y las bajas frecuencias son atenuadas por el factor  $\epsilon$ . Ahora, ya no hay polos en el origen, evitando así el fenómeno de deriva debido a la integración de ruidos del viento como (6.3).

<sup>1</sup>Haciendo  $|\mathbf{G}_{\text{HPF}_{df,x}}(j\omega)| = 1$ , se obtiene  $2a_{x_{o,4}}\omega^4 - 2a_{x_{o,2}}a_{x_{o,4}}\omega^2 + (1 + \epsilon)a_{x_{o,4}}^2 = 0$ .

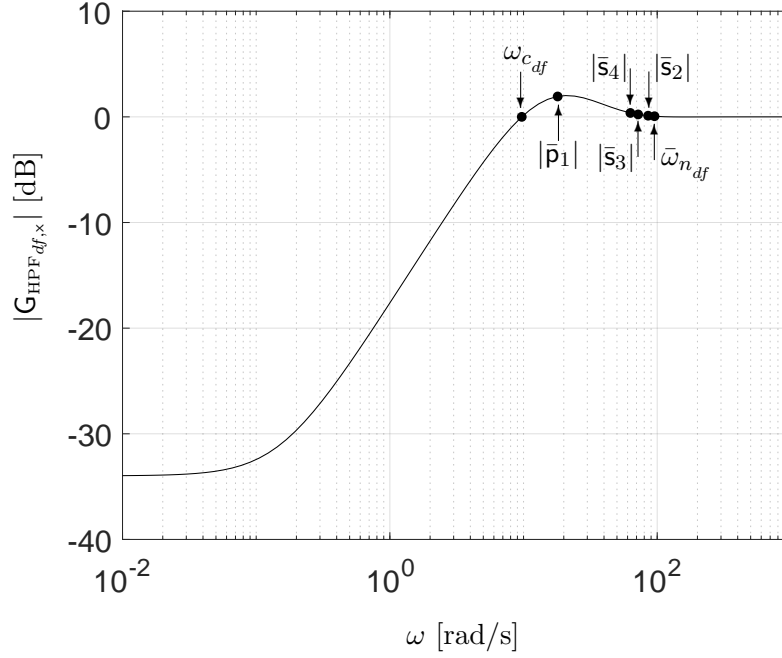


Figura 6.3: Gráfica de Bode de  $|G_{\text{HPF}_{df,x}}(j\omega)|_{dB}$ , 6.19 y 5.14 con  $\varrho_c = 18$  y  $\epsilon = 1/50$ .  $|\bar{p}_{1df}| = 18$ ,  $|\bar{s}_{2df}| = 85.5$ ,  $|\bar{s}_{3df}| = 72$ ,  $|\bar{s}_{4df}| = 63$ ,  $\bar{\omega}_{n_{df}} = 95.46$  rad/s,  $\bar{\zeta} = 0.966$  y  $\omega_{c_{df}} = 9.71$  rad/s;  $|G_{\text{HPF}_{df,x}}(0)| = \epsilon$  y  $|G_{\text{HPF}_{df,x}}(\infty)| = 1$ .

## 6.5 Seguimiento de trayectoria

En esta sección se presentan experimentos realizados en exteriores cuando el cuadricóptero sigue una trayectoria circular en el plano  $(x, y)$ . Para esto se procede de la siguiente manera

1. Primero se estabiliza localmente al cuadricóptero con el LQR de la sección 5.1, y es robustamente linealizado con los estimadores sin deriva de la sección 6.3.
2. Después se sintetiza una trayectoria de estado óptima para pasar de un punto estacionario local al siguiente punto estacionario local de la trayectoria circular.
3. Finalmente se particiona la trayectoria circular por un conjunto finito de puntos estacionarios locales adecuados.

### 6.5.1 Esquema de control

En la subsección 4.2.6, se mostró que el estimador de señal de incertidumbre no lineal (4.2) y (4.3) tiene dos propiedades interesantes

1. Agranda la vecindad de estabilidad (ver figura 4.16).
2. Crea un vecindad de atracción  $\mathcal{B}_{\rho_{**}}$ , alrededor del punto estacionario,  $x_i^*$ , (ver figura 4.14).

Entonces, las realimentaciones estabilizantes locales (5.4) y los estimadores sin deriva (6.8) y (6.9) toman las siguientes formas

$$u_i = F_i(x_i - x_i^*) + \bar{u}_{df,i}, \quad (6.21)$$

$$\begin{aligned} \frac{d}{dt} \bar{w}_{df,i} &= \left( \bar{A}_{\bar{K}_i} + (1 - \epsilon) \bar{B}_i G_i^\ell \bar{C}_i \right) \bar{w}_{df,i} - \left( \bar{K}_i + (1 - \epsilon) \bar{B}_i G_i^\ell \right) \bar{C}_i (x_i - x_i^*), \\ \bar{u}_{df,i} &= (1 - \epsilon) G_i^\ell (\bar{C}_i \bar{w}_{df,i} - \bar{C}_i (x_i - x_i^*)), \end{aligned} \quad (6.22)$$

donde  $i \in \{\mathbf{x}, \mathbf{y}\}$ , junto con los datos (5.6), (4.29), (4.31), (5.19) y  $\epsilon = 1/50$ .

### 6.5.2 Planificación de trayectoria de estado óptima

Se desea resolver el siguiente problema de control:

Sean los sistemas lineales descritos por las representaciones de estado (*cf.* (2.55)–(2.58), (5.4) y (5.6))

$$\frac{d}{dt} x_i^* = A_{F_i} x_i^* + B_i \bar{u}_i^*, \quad i \in \{\mathbf{x}, \mathbf{y}\}, \quad (6.23)$$

donde  $A_{F_i} \doteq A_i + B_i F_i$ ,  $i \in \{\mathbf{x}, \mathbf{y}\}$ , tal que los pares  $(A_{F_i}, B_i)$  son controlables (*cf.* (2.56) y (2.58)).

Sea una partición de la trayectoria deseada  $x_i^*(t)$  ( $i \in \{\mathbf{x}, \mathbf{y}\}$ ),  $t \in [0, T_f]$ , en  $N + 1$  puntos estacionarios, es decir

$$\Lambda_{SP_i}^* = \{ \bar{x}_{0,i}^*, \bar{x}_{1,i}^*, \dots, \bar{x}_{N,i}^* \}, \quad x_i^*(kT_s) = \bar{x}_{k,i}^*, \quad k \in \{0, 1, \dots, N\}, \quad (6.24)$$

donde  $NT_s = T_f$  y  $i \in \{\mathbf{x}, \mathbf{y}\}$ ;  $T_s$  es el tiempo de muestreo de la trayectoria y  $T_f$  es el tiempo de vuelo.

Encontrar entradas de control de norma mínima  $\bar{u}_i^*$ , tales que las trayectorias solución, partiendo del punto estacionario  $x_i^*(kT_s) = x_{k,i}$ , alcancen al siguiente punto estacionario en tiempo finito, esto es:  $x_i^*((k+1)T_s) = \bar{x}_{(k+1),i}^*$ , donde  $T_s$  es el tiempo de muestreo dado,  $T_s > 0$ .

Este es un problema clásico de norma mínima de buscar el vector más cercano al origen que se encuentra en una variedad de codimensión finita, en un espacio de Hilbert, y se resuelve con la ayuda del teorema de la proyección.

De acuerdo al teorema 2 de la sección 3.3 de [40], la entrada de control  $\bar{u}_i^*$  para resolver el problema tiene la forma (ver también el teorema 1.1 de [26])

$$\bar{u}_i^*(t) = -\mathcal{F}_{F_i}(T_s - t) \mathfrak{W}_{i_{T_s}}^{-1} \exp(A_{F_i} T_s) x_{o,i}, \quad i \in \{\mathbf{x}, \mathbf{y}\}, \quad t \in [0, T_s], \quad (6.25)$$

donde<sup>2</sup>

$$\mathcal{F}_{F_i}(t) = B_i^T \exp(A_{F_i}^T t), \quad \mathfrak{W}_{i_{T_s}} = \int_0^{T_s} \mathcal{F}_{F_i}^T(T_s - \tau) \mathcal{F}_{F_i}(T_s - \tau) d\tau. \quad (6.26)$$

De (6.23), (6.25) y (6.26), obtenemos para todos los intervalos de tiempo,  $[kT_s, (k+1)T_s)$ ,  $k \in \{0, 1, \dots, (N-1)\}$ , las siguientes trayectorias óptimas

$$x_i^*(t) = \exp(A_{F_i}(t - kT_s)) \bar{x}_{k,i}^* + \int_{kT_s}^t \mathcal{F}_{F_i}^T(t - \tau) \beta_i^*(t, \tau) v_{k,i}^* d\tau, \quad (6.27)$$

donde (recordar (6.24))

$$\begin{aligned} \beta_i^*(t, \tau) &= \mathcal{F}_{F_i}(t - \tau) \mathfrak{W}_{i_{T_s}}^{-1}, \\ v_{k,i}^* &= \bar{x}_{(k+1),i}^* - \exp(A_{F_i} T_s) \bar{x}_{k,i}^*, \\ \bar{x}_{k,i}^* &\in \Lambda_{SP_i}^*, \quad i \in \{\mathbf{x}, \mathbf{y}\} \end{aligned} \quad (6.28)$$

En efecto, al instante  $t = kT_s$  los sistemas (6.23) tienen las condiciones iniciales  $x_i^*(kT_s) = \bar{x}_{k,i}^*$ . En el intervalo de tiempo  $[kT_s, (k+1)T_s)$  se aplican las entradas de control (6.25) y (6.26), de norma mínima. Entonces al instante  $t = (k+1)T_s$ , las trayectorias solución alcanzan los siguientes puntos estacionarios,  $x_i^*((k+1)T_s) = \bar{x}_{(k+1),i}^*$  (cf. (6.27) y (6.24)).

---

<sup>2</sup>Como los pares  $(A_{F_i}, B_i)$  son controlables, las matrices  $\mathfrak{W}_{i_{T_s}}$  son invertibles [26].



### 6.5.3 Partición de la trayectoria circular

Estamos interesados en seguir la trayectoria circular

$$x_*^2(t) + (y_*(t) - r_*)^2 = r_*^2, \quad (6.29)$$

esto es

$$\begin{aligned} x_*(t) &= r_* \sin(\omega_s t), \\ y_*(t) &= r_* (1 - \cos(\omega_s t)), \end{aligned} \quad (6.30)$$

donde  $\omega_s = 2\pi/T_f$ .

De (2.56), (2.58) y (6.23), tenemos

$$\begin{aligned} x_{1,x}^* &= r_* \sin(\omega_s t), \\ x_{1,y}^* &= r_* (1 - \cos(\omega_s t)), \end{aligned} \quad (6.31)$$

$$\begin{aligned} x_{2,x}^* &= \frac{d}{dt} x_{1,x}^* = r_* \omega_s \cos(\omega_s t), \\ x_{2,y}^* &= \frac{d}{dt} x_{1,y}^* = r_* \omega_s \sin(\omega_s t), \end{aligned} \quad (6.32)$$

$$\begin{aligned} x_{3,x}^* &= -g^{-1} \frac{d}{dt} x_{2,x}^* = g^{-1} r_* \omega_s^2 \sin(\omega_s t), \\ x_{3,y}^* &= g^{-1} \frac{d}{dt} x_{2,y}^* = g^{-1} r_* \omega_s^2 \cos(\omega_s t), \end{aligned} \quad (6.33)$$

$$\begin{aligned} x_{4,x}^* &= \frac{d}{dt} x_{3,x}^* = g^{-1} r_* \omega_s^3 \cos(\omega_s t), \\ x_{4,y}^* &= \frac{d}{dt} x_{3,y}^* = -g^{-1} r_* \omega_s^3 \sin(\omega_s t). \end{aligned} \quad (6.34)$$

Entonces, particionando  $T_f$  en  $N$  puntos, de (6.31)–(6.34) obtenemos ( $t = (k/N)T_f$ ,  $k \in \{0, 1, \dots, N\}$ )

$$\bar{x}_{k,x}^* = r_* \left[ \sin(\alpha_k) \quad \omega_s \cos(\alpha_k) \quad g^{-1} \omega_s^2 \sin(\alpha_k) \quad g^{-1} \omega_s^3 \cos(\alpha_k) \right]^T, \quad (6.35)$$

$$\bar{x}_{k,y}^* = r_* \left[ (1 - \cos(\alpha_k)) \quad \omega_s \sin(\alpha_k) \quad g^{-1} \omega_s^2 \cos(\alpha_k) \quad -g^{-1} \omega_s^3 \sin(\alpha_k) \right]^T, \quad (6.36)$$

donde  $\alpha_k = 2\pi k/N$ .

## 6.5. Seguimiento de trayectoria

En las figuras 6.4 y 6.5, a modo de ejemplo, se muestran resultados de simulación, para una partición gruesa, cada  $9^\circ$ . Se utilizaron los datos de la sección 4.2.

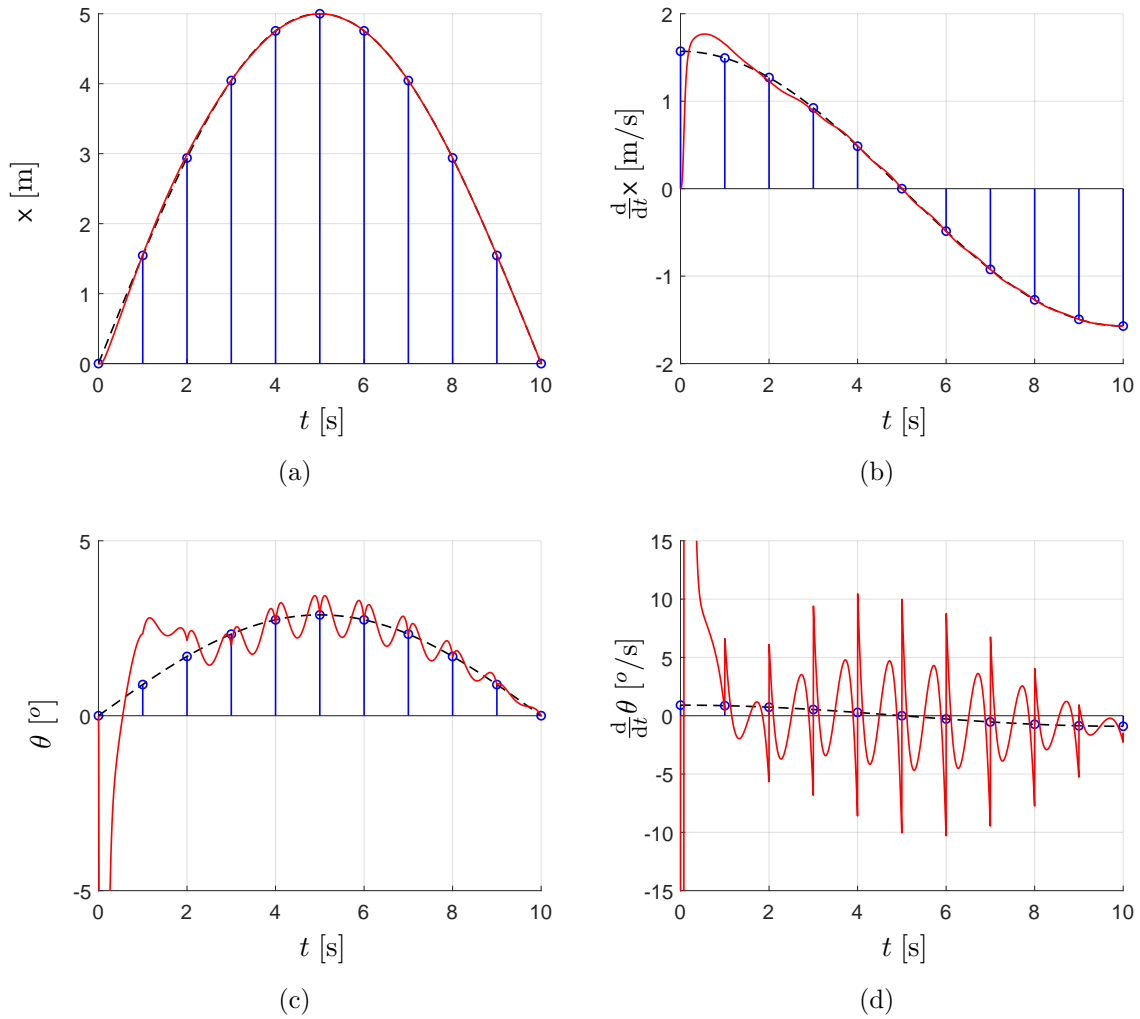


Figura 6.4: Simulación numérica para una partición gruesa, cada  $9^\circ$ , utilizando los datos de la sección 4.2. En las figuras (a), (b), (c) y (d) se muestran las trayectorias deseadas (línea discontinua), la trayectoria seguida por el cuadricóptero (color rojo) y los puntos estacionarios (color azul).

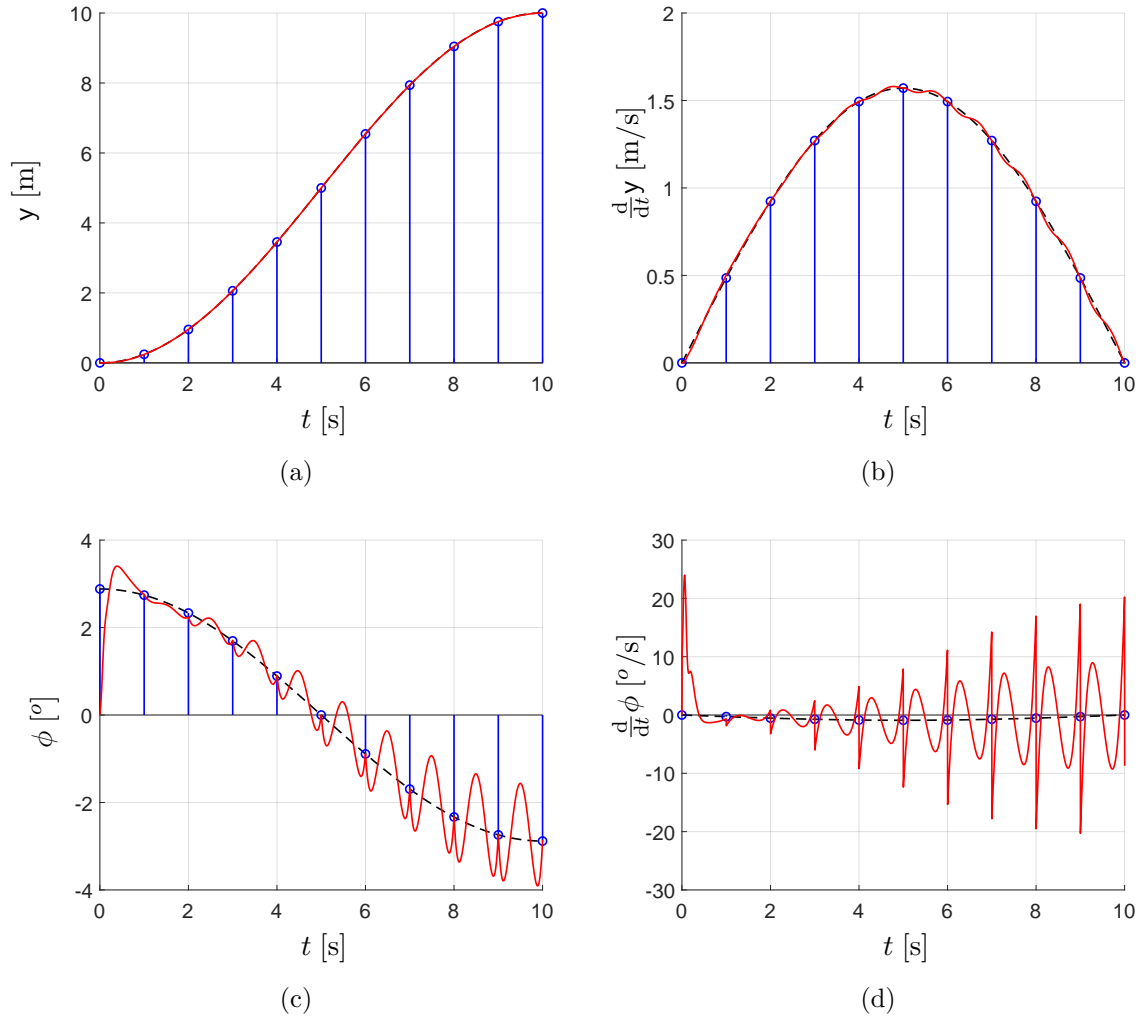


Figura 6.5: Simulación numérica para una partición gruesa, cada  $9^\circ$ , utilizando los datos de la sección 4.2. En las figuras (a), (b), (c) y (d) se muestran las trayectorias deseadas (línea discontinua), la trayectoria seguida por el cuadricóptero (color rojo) y los puntos estacionarios (color azul).

Para sintetizar la trayectoria de estado óptima dada por (6.27) y (6.28), primero se ha aplicado el cambio de base  $(\bar{A}_{F_i}, \bar{B}_i) = (T_i^{-1}A_{F_i}T_i, T_i^{-1}B_i)$ ,  $i \in \{x, x\}$ , para simplificar el

cálculo de la matriz exponencial, donde

$$T_x^{-1} = \begin{bmatrix} 0 & 626.5 & -3216.0 & 0 \\ 0 & 1.205 & -0.003678 & 0 \\ 1 & 0.5236 & -0.001597 & 0 \\ 0 & -626.1 & 3216.0 & 1 \end{bmatrix} \quad \text{y} \quad T_y^{-1} = \begin{bmatrix} 0 & 654.5 & 3360.0 & 0 \\ 0 & 1.205 & 0.00352 & 0 \\ 1 & 0.5236 & 0.001529 & 0 \\ 0 & 654.2 & 3360 & 1 \end{bmatrix} \quad (6.37)$$

Las integrales de convolución (6.27) fueron sintetizadas con la discretización de la descripción en espacio de estados  $\Sigma(\bar{A}_{F_i}, \bar{B}_i)$ ,  $i \in \{x, y\}$ , en el intervalo de tiempo  $[0, T_s)$ ,  $T_s = 1$ , y con las condiciones iniciales (6.35) y (6.36).

Las matrices exponenciales de (6.27) son

$$\begin{aligned} \exp(A_{F_x} T_s) &= T_x \exp(\bar{A}_{F_x} T_s) T_x^{-1} = \begin{bmatrix} 0.6118 & 0.4581 & -1.1060 & -0.0003 \\ -0.7364 & -0.1841 & -0.2737 & -0.0001 \\ 0.0186 & 0.0952 & -0.2909 & -0.0001 \\ -0.1938 & -0.1908 & 0.1764 & 0.0001 \end{bmatrix}, \\ \exp(A_{F_y} T_s) &= T_y \exp(\bar{A}_{F_y} T_s) T_y^{-1} = \begin{bmatrix} 0.6118 & 0.4581 & 1.1060 & 0.0003 \\ -0.7364 & -0.1841 & 0.2737 & 0.0001 \\ -0.0186 & -0.0952 & -0.2909 & -0.0001 \\ 0.1938 & 0.1908 & 0.1764 & 0.0001 \end{bmatrix}. \end{aligned} \quad (6.38)$$

Las  $\beta_i^*(t, \tau)$  son

$$\begin{aligned} \beta_x^*(t, \tau) &= \mathcal{F}_{F_x}(t - \tau) \mathfrak{W}_{xT_s}^{-1} T_x^{-1} \\ &= k_{1,x} e^{a_{1,x}(t-1)} + k_{2,x} e^{a_{2,x}(t-1)} + (k_{3,x} \cos(\omega_x(t-1)) + k_{4,x} \sin(\omega_x(t-1))) e^{b_x(t-1)} \end{aligned} \quad (6.39)$$

$$\begin{aligned} \beta_y^*(t, \tau) &= \mathcal{F}_{F_y}(t - \tau) \mathfrak{W}_{yT_s}^{-1} T_y^{-1} \\ &= k_{1,y} e^{a_{1,y}(t-1)} + k_{2,y} e^{a_{2,y}(t-1)} + (k_{3,y} \cos(\omega_y(t-1)) + k_{4,y} \sin(\omega_y(t-1))) e^{b_y(t-1)} \end{aligned} \quad (6.40)$$

donde  $a_{1,x} = 3214$ ,  $a_{2,x} = 1.911$ ,  $\omega_x = 1.585$ ,  $b_x = 0.9536$ ,  $a_{1,y} = 3358$ ,  $a_{2,y} = 1.911$ ,  $\omega_y = 1.585$ ,  $b_y = 0.9536$ ,

$$\begin{aligned}
 [k_{1,x} \ k_{2,x} \ k_{3,x} \ k_{4,x}] &= \begin{bmatrix} 556.9 & -2266.0 & 1957.0 & -182.2 \\ -333.6 & 1155.0 & -1020.0 & 171.6 \\ -817.3 & 2234.0 & -1652.0 & 531.3 \\ 90.36 & -0.6939 & 0.5131 & -0.1651 \end{bmatrix}, \\
 [k_{1,y} \ k_{2,y} \ k_{3,y} \ k_{4,y}] &= \begin{bmatrix} -556.8 & 2266.0 & -1957.0 & 182.1 \\ 333.5 & -1154.0 & 1020.0 & -171.6 \\ -817.2 & 2234.0 & -1652.0 & 531.2 \\ 90.35 & -0.6641 & 0.4911 & -0.158 \end{bmatrix}.
 \end{aligned}$$

## 6.6 Experimentos en exteriores

En esta sección presentamos los experimentos realizados en exteriores, cuando seguimos una trayectoria circular de radio  $r_* = 5$  m, en el plano  $(x, y)$ . Para esto, seguimos los pasos indicados en la sección 6.5, es decir

- Se estabiliza localmente el cuadricóptero con las realimentaciones de estado (6.21) (para  $i \in \{x, y, \psi\}$ ) y (5.5) (para  $i = z$ ), junto con (5.6).
- Se linealiza robustamente el cuadricóptero con los estimadores sin deriva (6.22) (para  $i \in \{x, y, \psi\}$ ), junto con (4.29), (4.31), (5.19) y  $\epsilon = 1/50$ .
- Se genera la trayectoria circular para el seguimiento con la ayuda de (6.27), (6.28), (6.26), (6.35) y (6.36), con  $r_* = 5$  m y  $T_s = 1$  s (recordar (2.56), (2.58) y (5.6)).

Se hicieron dos experimentos

1. Se siguió una trayectoria circular de  $r_* = 5$  m de radio, en un tiempo de 60 s.
2. Se siguió una trayectoria circular de  $r_* = 5$  m de radio, en un tiempo de 20 s.

### 6.6.1 Trayectoria lenta

Primero se siguió la trayectoria circular (6.29) con radio  $r_* = 5$  m, tiempo de muestreo  $T_s = 1$  s y una partición de  $N = 60$  puntos, esto es:  $T_f = 60$  s.

En las figuras 6.6 y 6.7 se muestran los resultados obtenidos cuando se aplican las realimentaciones localmente estabilizantes (6.21), con y sin los estimadores sin deriva (6.22). En las figuras 6.6a y 6.6b se comparan la trayectoria obtenida con la trayectoria objetivo (6.29). En las figuras 6.6c y 6.6d se comparan las trayectorias  $(x, y)$  del cuadricóptero con las referencias (6.30). En las figuras 6.7a y 6.7b se muestra el error de seguimiento  $e_c(t)$

$$e_c(t) = \sqrt{(x(t) - x_*(t))^2 + (y(t) - y_*(t))^2}. \quad (6.41)$$

Cuando se aplican las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22),  $i \in \{x, y\}$ , obtenemos el pico de error máximo y el error cuadrático medio

$$\|e_c\|_p = 0.3046 \text{ m} \quad \text{y} \quad \|e_c\|_{rms} = 0.1287 \text{ m}, \quad (6.42)$$

donde<sup>3</sup>

$$\|e_c\|_p = \max_{t \in [0, T_f]} e_c(t), \quad \|e_c\|_{rms} = \sqrt{\frac{1}{T_f} \int_0^{T_f} e_c^2(t) dt}. \quad (6.43)$$

Cuando se aplican solamente las realimentaciones localmente estabilizantes (6.21),  $i \in \{x, y\}$ , obtenemos el pico de error máximo y el error cuadrático medio

$$\|e_c\|_p = 0.9493 \text{ m} \quad \text{y} \quad \|e_c\|_{rms} = 0.5534 \text{ m}. \quad (6.44)$$

Por lo tanto, con la combinación de las realimentaciones localmente estabilizantes (6.21) y los estimadores sin deriva (6.22), obtenemos las eficiencias

$$\epsilon_{ffp} = 93.908 \% \quad \text{y} \quad \epsilon_{ffrms} = 96.360 \%, \quad (6.45)$$

---

<sup>3</sup>Recordar que para el caso de señales sinusoidales el error cuadrático medio es:  $\sqrt{(1/T_f) \int_0^{T_f} \sin^2(2\pi t/T_f) dt} = 1/\sqrt{2}$ .

y solamente con las realimentaciones localmente estabilizantes (6.21), obtenemos las eficiencias

$$\boldsymbol{\epsilon}_{\text{ff}_p} = 81.014\% \quad \text{y} \quad \boldsymbol{\epsilon}_{\text{ff}_{rms}} = 84.384\%, \quad (6.46)$$

donde

$$\boldsymbol{\epsilon}_{\text{ff}_p} = \left(1 - \|e_c\|_p / \mathbf{r}_*\right) \times 100\%, \quad \boldsymbol{\epsilon}_{\text{ff}_{rms}} = \left(1 - \sqrt{2} \|e_c\|_{rms} / \mathbf{r}_*\right) \times 100\%. \quad (6.47)$$

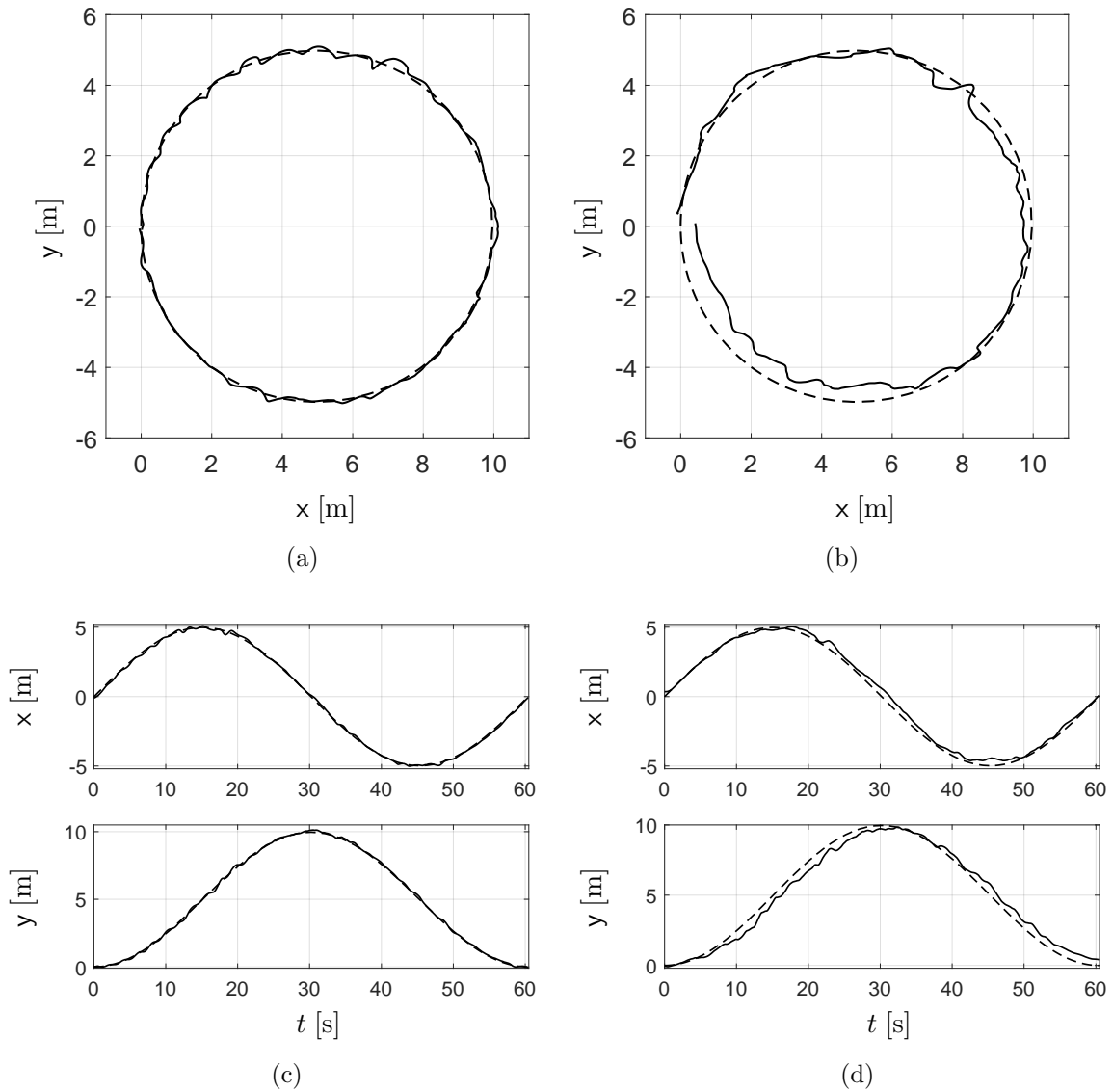


Figura 6.6: Seguimiento de una trayectoria circular en el plano  $(x, y)$  con 5 m de radio en 60 s ( $N = 60$  y  $T_s = 1$  s). (a) y (c) Aplicación de las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22). (b) y (d) Aplicando solamente las realimentaciones localmente estabilizantes (6.21). (a) y (b) Comparación de la trayectoria obtenida con la trayectoria objetivo (6.29). (c) y (d) Comparación de las trayectorias en  $(x, y)$  con las referencias (6.30).



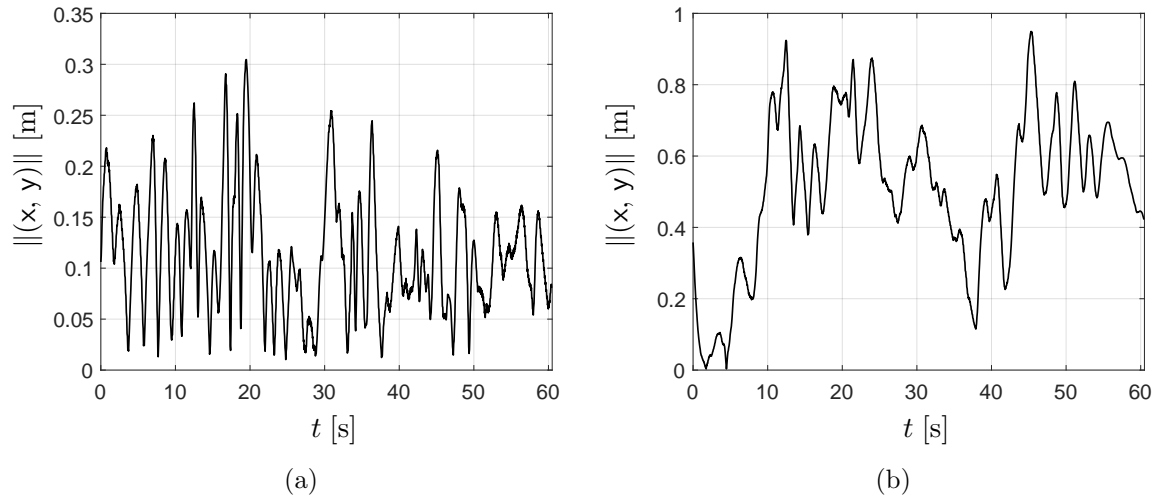


Figura 6.7: Comparación del error de seguimiento  $e_c(t)$  (ver (6.41)). (a) Aplicando las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22). (b) Aplicando solamente las realimentaciones localmente estabilizantes (6.21).

### 6.6.2 Trayectoria rápida

A continuación se siguió la trayectoria circular (6.29) con radio  $r_* = 5$  m, tiempo de muestreo  $T_s = 1$  s y una partición de  $N = 20$  puntos, esto es:  $T_f = 20$  s.

En las figuras 6.8 y 6.9 se muestran los resultados obtenidos cuando se aplican las realimentaciones localmente estabilizantes (6.21), con y sin los estimadores sin deriva (6.22). En las figuras 6.8a y 6.8b se comparan la trayectoria obtenida con la trayectoria objetivo (6.29). En las figuras 6.8c y 6.8d se comparan las trayectorias (x, y) del cuadricóptero con las referencias (6.30). En las figuras 6.9a y 6.9b se muestra el error de seguimiento  $e_c(t)$ .

Cuando se aplican las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22),  $i \in \{x, y\}$ , obtenemos el pico de error máximo y el error cuadrático medio

$$\|e_c\|_p = 0.5463 \text{ m} \quad \text{y} \quad \|e_c\|_{rms} = 0.2992 \text{ m}. \quad (6.48)$$

Mientras que cuando se aplican solamente las realimentaciones localmente estabilizantes (6.21),  $i \in \{x, y\}$ , obtenemos el pico de error máximo y el error cuadrático medio

$$\|e_c\|_p = 2.3582 \text{ m} \quad \text{y} \quad \|e_c\|_{rms} = 1.0277 \text{ m}. \quad (6.49)$$

Con la combinación de las realimentaciones localmente estabilizantes (6.21) y los estimadores sin deriva (6.22), obtenemos las eficiencias

$$\epsilon_{ff_p} = 89.074 \% \quad \text{y} \quad \epsilon_{ff_{rms}} = 91.537 \%, \quad (6.50)$$

y solamente con las realimentaciones localmente estabilizantes (6.21), obtenemos las eficiencias

$$\epsilon_{ff_p} = 52.836 \% \quad \text{y} \quad \epsilon_{ff_{rms}} = 70.932 \%, \quad (6.51)$$

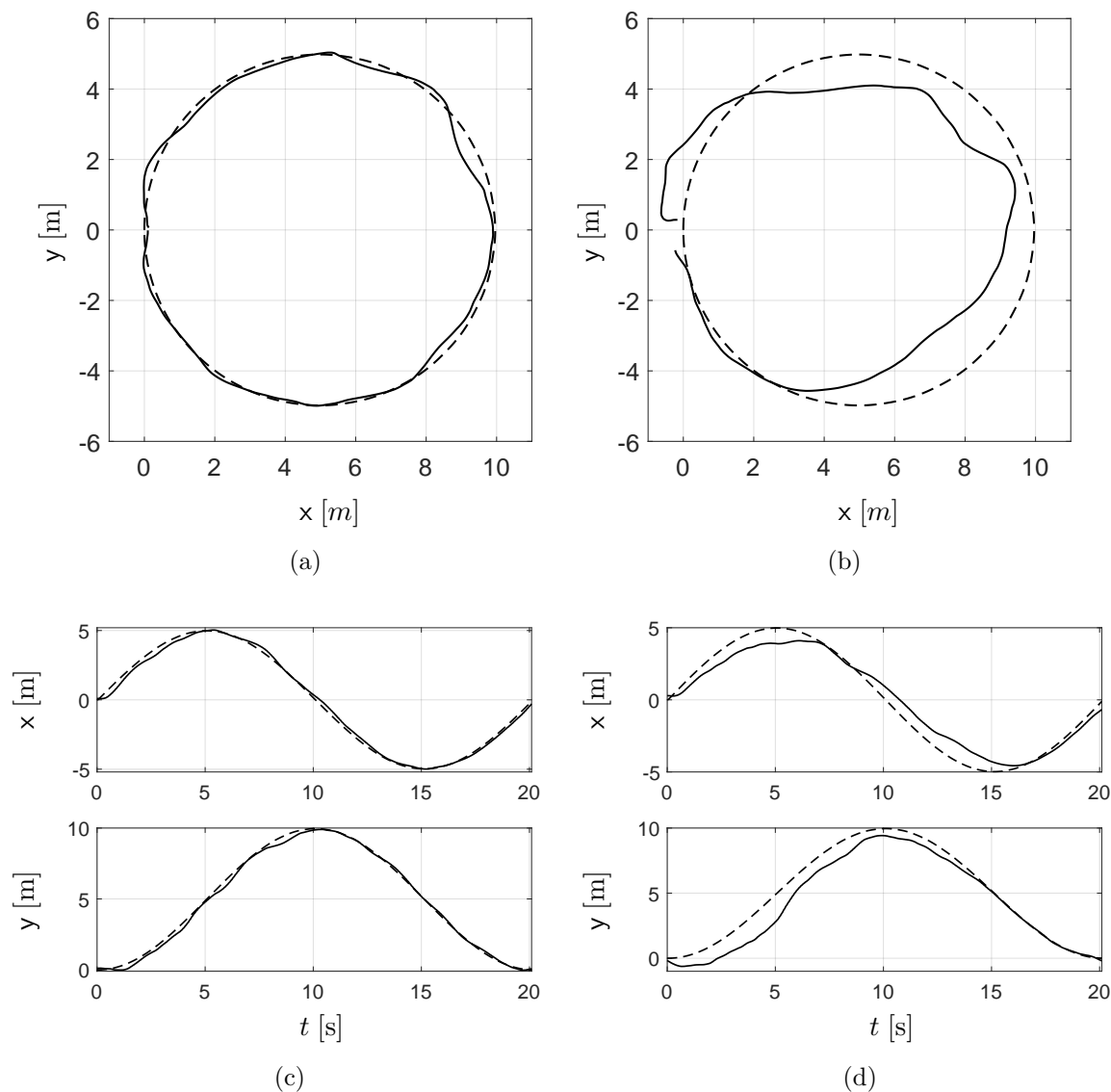


Figura 6.8: Seguimiento de una trayectoria circular en el plano  $(x, y)$  con 5 m de radio en 20 s ( $N = 20$  y  $T_s = 1$  s). (a) y (c) Aplicación de las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22). (b) y (d) Aplicando solamente las realimentaciones localmente estabilizantes (6.21). (a) y (b) Comparación de la trayectoria obtenida con la trayectoria objetivo (6.29). (c) y (d) Comparación de las trayectorias en  $(x, y)$  con las referencias (6.30).

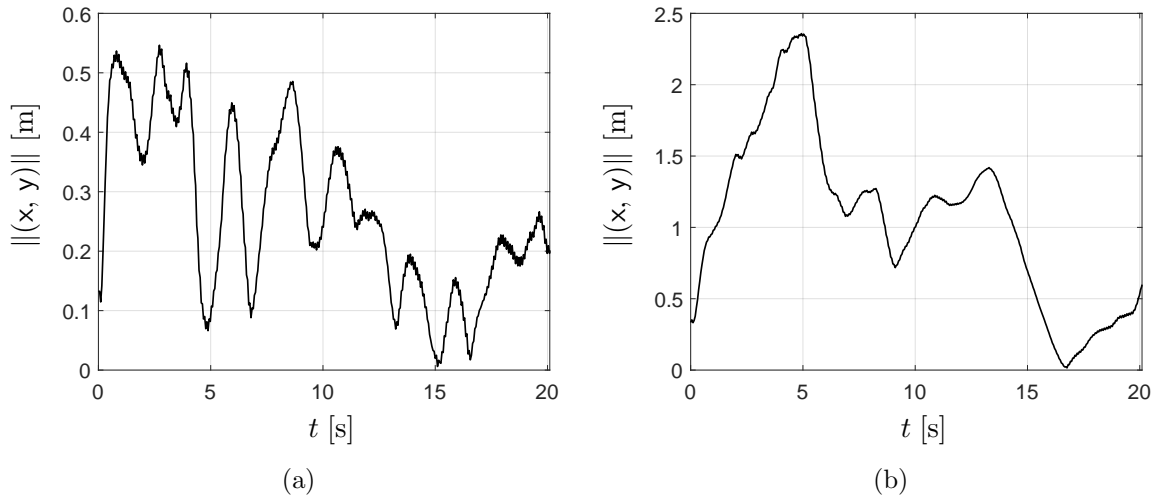


Figura 6.9: Comparación del error de seguimiento  $e_c(t)$  (ver (6.41)). (a) Aplicando las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22). (b) Aplicando solamente las realimentaciones localmente estabilizantes (6.21).

El cuadricóptero se voló a una altura de alrededor de 3 y 6 metros, ver figuras 6.10a, 6.10b, 6.11a y 6.11b. En las figuras 6.12 y 6.13 se muestran las acciones de control  $u_x$ ,  $u_y$ ,  $u_z$  y  $u_\psi$  para la trayectoria rápida:  $N = 20$  y  $T_s = 1$  s (recordar (6.21), (5.5), (5.6) y (6.22)).

A partir de estos experimentos realizados en exteriores, vemos que para la trayectoria lenta obtenemos una mejora de la eficiencia de alrededor del 10%, cuando se aplican los estimadores sin deriva (6.22); y para la trayectoria rápida la mejora de la eficiencia fue de alrededor del 20%. Observamos que error en guiñada disminuye en un factor de 1/3 cuando se aplica el estimador si deriva (6.22); ver figuras 6.10 y 6.11.

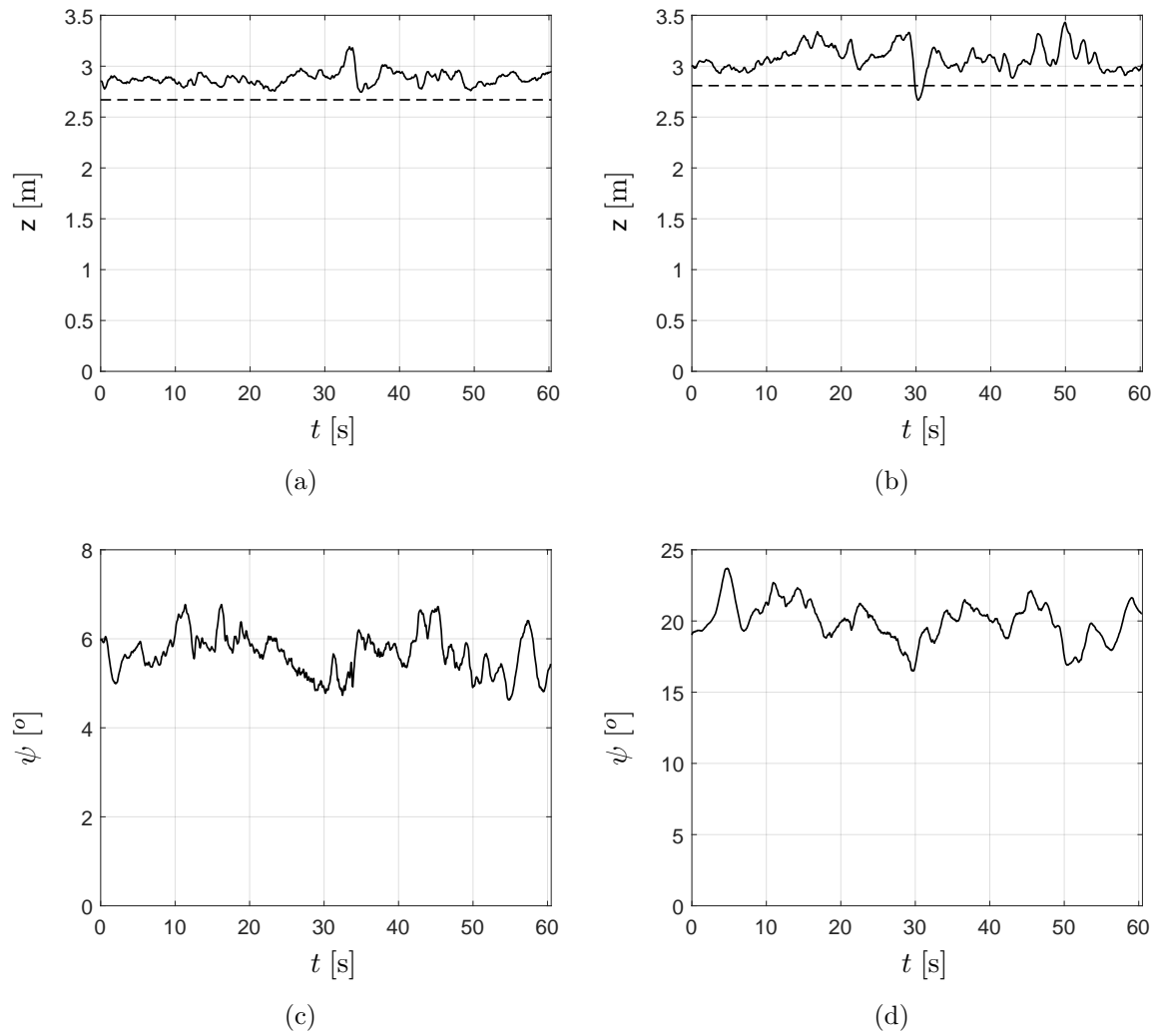


Figura 6.10: Altura  $z$  y ángulo de guiñada  $\psi$ . (a) y (b) Aplicando únicamente (5.5). (c) Aplicando (6.21) y (6.22). (d) Aplicando únicamente (6.21). (a)-(d) Seguimiento lento.

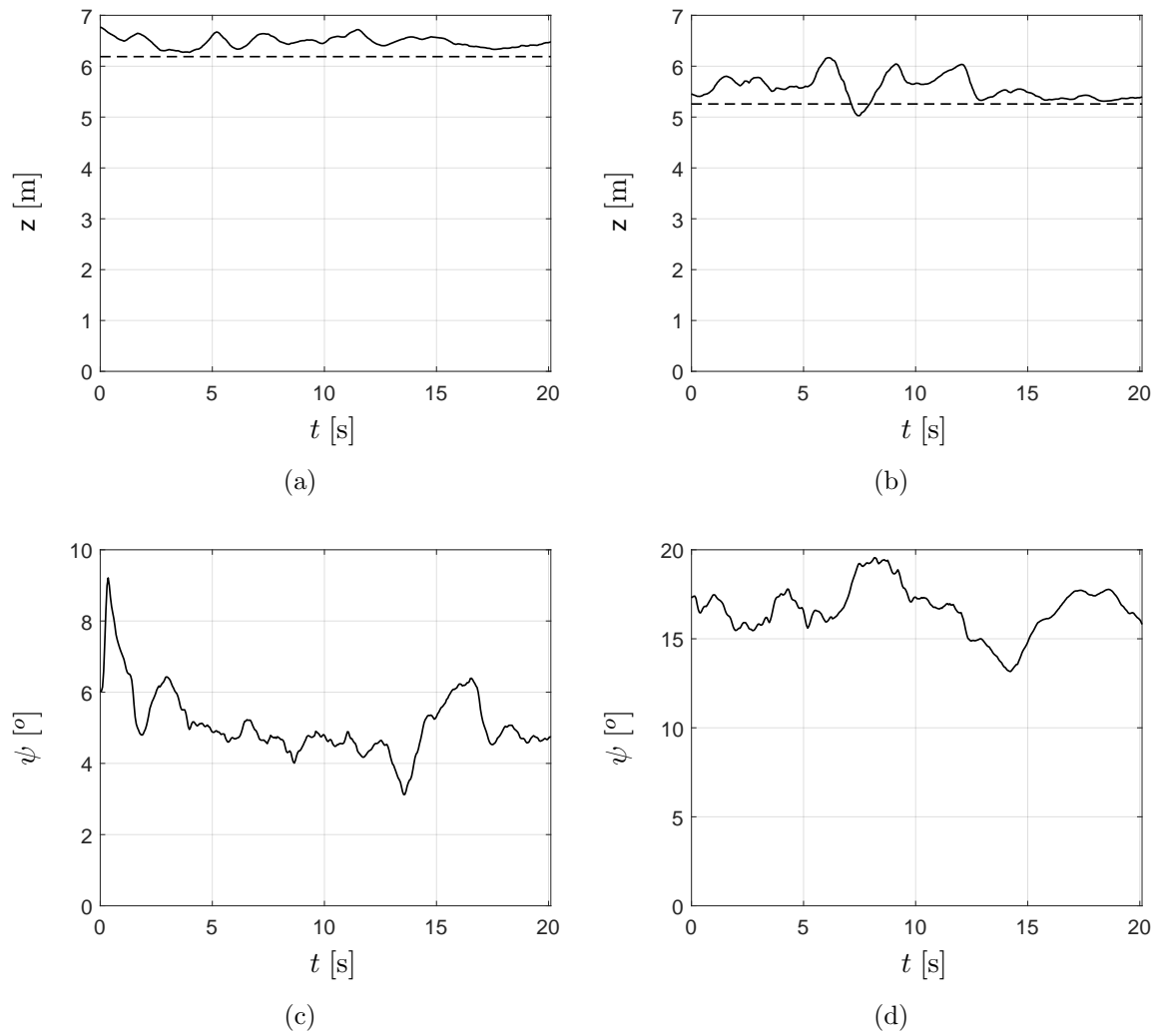


Figura 6.11: Altura  $z$  y ángulo de guiñada  $\psi$ . (a) y (b) Aplicando únicamente (5.5). (c) Aplicando (6.21) y (6.22). (d) Aplicando únicamente (6.21). (a)-(d) Seguimiento rápido.

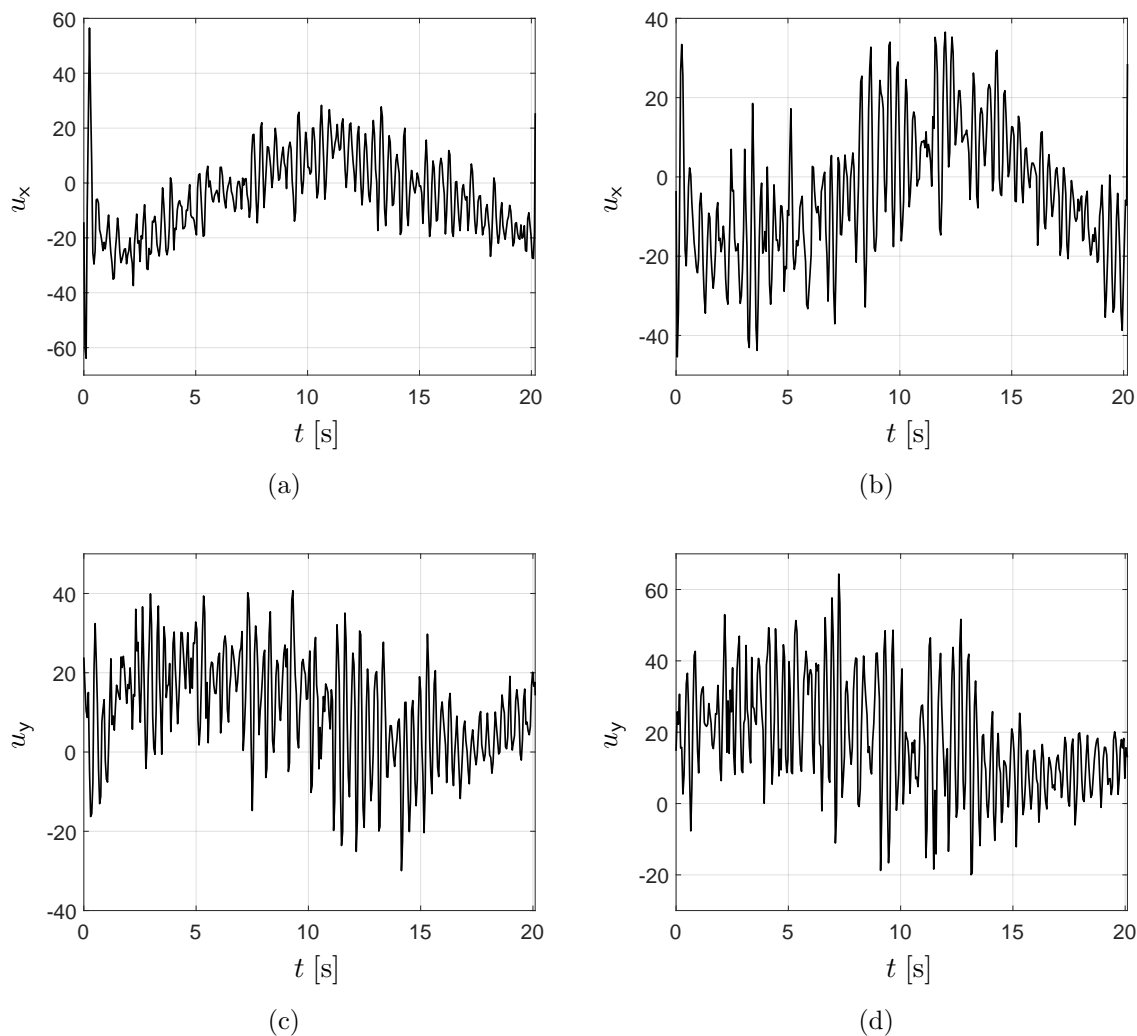


Figura 6.12: Acciones de control para el seguimiento rápido de trayectoria. (a) y (c) Aplicando las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22). (b) y (d) Aplicando solamente las realimentaciones localmente estabilizantes (6.21).

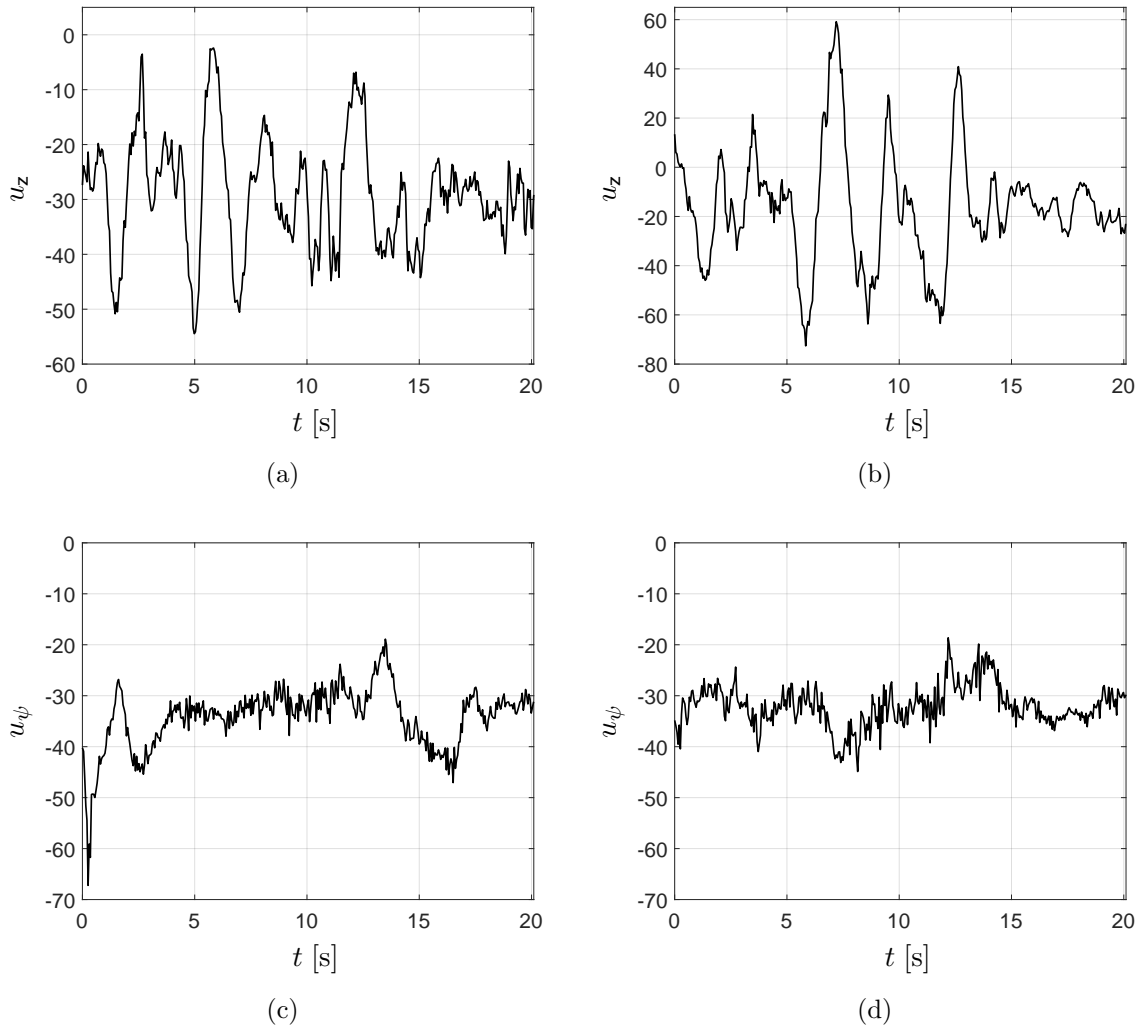


Figura 6.13: Acciones de control para el seguimiento rápido de trayectoria. (a) y (b) Aplicando únicamente (5.5). (c) Aplicando las realimentaciones localmente estabilizantes (6.21) junto con los estimadores sin deriva (6.22). (d) Aplicando solamente las realimentaciones localmente estabilizantes (6.21).

## 6.7 Conclusiones

En este capítulo se abordó el problema de seguimiento de trayectoria en el plano  $(x, y)$ . Se encontró que existe un fenómeno de deriva debido al ruido del viento, este se caracterizó



mediante la expresión matemática (6.2) y se comprobó que cuando el término (6.3) pasa a través de la acción integral del estimador de incertidumbre no lineal, ocurre el fenómeno de deriva. Este inconveniente se evitó al desplazar el polo del origen de (4.8), ligeramente a la izquierda del plano complejo, lo anterior da lugar a el estimador sin deriva (6.6).

Se sintetizó una trayectoria de estado óptima para pasar de un punto estacionario local al siguiente punto estacionario local de una trayectoria circular. Esta trayectoria se particionó en un conjunto finito de puntos estacionarios locales.

Se presentaron resultados experimentales realizados en exteriores, cuando el cuadricóptero sigue una trayectoria circular de radio  $r_* = 5$  m en el plano  $(x,y)$ . Primero se siguió una trayectoria circular de 5 m de radio en 60 s (trayectoria lenta) y luego una trayectoria circular de 5 m de radio en 20 s (trayectoria rápida). Se obtuvo una reducción en los errores de seguimiento y guiñada al aplicar los estimadores sin deriva (6.22). En la trayectoria lenta se obtuvo una mejora en la eficiencia de alrededor del 10 % y en la trayectoria rápida una mejora en la eficiencia de alrededor del 20 %; con respecto al error en guiñada, este disminuyó en un factor de 1/3.

# Capítulo 7

## Conclusiones

Es este trabajo de tesis se desarrolló e implementó un esquema de control lineal capaz de rechazar las no linealidades del modelo y compensar los cambios inesperados de la estructura.

En la sección 3.2 se mostró que un sistema no lineal modelado por la representación de estado (3.18) puede transformarse a la representación de estado (3.30). La diferencia entre las representaciones de estado (3.18) y (3.30) radica en la manera que actúan las no linealidades en las representaciones de estado. En (3.18) las no linealidades están caracterizadas por el vector de señal de perturbación no lineal  $q$ , el cual actúa a través del mapa  $S$ . En (3.30) las no linealidades están caracterizadas por el vector de señal de incertidumbre no lineal  $q_*$ , el cual actúa directamente en la imagen de  $B$ . Esta transformación se logra por el cambio de variable (3.29), la cual existe bajo condiciones de controlabilidad del par  $(A, B)$ . Dado que ahora la señal de incertidumbre no lineal  $q_*$  (3.31) se encuentra dentro de la imagen de  $B$ , entonces se puede cancelar directamente mediante la entrada de control  $u$ .

Esta linealización exacta se basa en la reconstrucción analítica de  $q_*$ ; pero si esto no es posible o es una tarea pesada y tediosa hacerlo, se puede estimar. Para esto se propuso el estimador de señal de incertidumbre no lineal (4.2) basado en el filtro Beard-Jones, cuyo

objetivo es rechazar robustamente la señal de incertidumbre no lineal  $q_*$ .

La simplicidad y efectividad de (4.2) es especialmente adecuada para vehículos aéreos no tripulados, donde es importante no solo rechazar las no linealidades y los cambios inesperados de la estructura, sino también buscar la simplicidad y efectividad del esquema de control.

También se mostró que la función de transferencia (4.16) del sistema en lazo cerrado (4.13) tiene incorporado el filtro pasa altas  $1 - F_e(s)$  cuya finalidad es rechazar las frecuencias bajas de  $q_*$  (recordar figura 4.1). En la subsección 4.2.5 se mencionó que la frecuencia de corte  $\omega_c$  del filtro pasa bajas  $F_e(s)$  debe ser suficientemente mayor que el ancho de banda  $\mathcal{B}_W$  de la transformada de Fourier de  $\mathcal{F}\{q_*\}$ . En la subsección 4.2.6, se mostró que otra característica importante del filtro Beard-Jones (4.2) es que agranda la vecindad de estabilidad y produce una vecindad de atracción (ver teorema 4.2.1 y colorario 4.2.1).

En el capítulo 5, se mostró como sintetizar la linealización robusta asintótica por realimentación, (4.2) y (5.4), para el cuadricóptero en vuelo estacionario. Para esto, primero se estabilizó localmente a (2.55)-(2.62) usando la técnica LQR. Para sintetizar los estimadores de señal de incertidumbre no lineal (4.28)-(4.31), en la sección 5.2 se mostró que la frecuencia de corte del filtro pasa altas  $1 - F_e(s)$  depende solo de la constante de parametrización  $\varrho_c$ , por lo tanto (ver (5.18)),  $\varrho_c$  tiene que ajustarse suficientemente más grande que el ancho de banda  $\mathcal{B}_W$  de  $\mathcal{F}\{q_{i,*}\}$  (ver figura 5.1). Se mostró que el ancho de banda  $\mathcal{B}_W$  de  $q_{i,*}$  puede estimarse con la ayuda de la transformada rápida de Fourier de datos experimentales de vuelo estacionario (ver figura 5.2).

En la sección 5.3 se presentaron resultados experimentales de vuelo estacionario en exteriores, considerando dos casos: cuando aplican únicamente realimentaciones de estado y realimentaciones de estado más el estimador de incertidumbre. En los resultados comparativos puede notarse que el esquema de control propuesto tiene mejor desempeño que solamente

---

aplicar las realimentaciones de estado. Debido a la acción del alargamiento de la vecindad de estabilidad y la vecindad de atracción del estimador de incertidumbre no lineal (4.28)–(4.31) (ver figura 5.4).

En el capítulo 6, se atacó el problema de seguimiento de trayectoria. Para este caso se encontró que existe un fenómeno de deriva debido al ruido del viento el cual causa problemas de deriva. Este ruido se caracterizó mediante la expresión (6.2). En la sección 6.2 se comprobó que cuando el término (6.3) pasa a través de la acción integral del estimador de incertidumbre no lineal, ocurre el fenómeno de deriva. Para superar el fenómeno de deriva del cuadricóptero, en la sección 6.3 se propuso desplazar ligeramente a la izquierda del plano complejo al polo en el origen de (4.8), obteniéndose el estimador sin deriva (6.6) (ver (6.7)).

En la sección 6.4 se caracterizó la respuesta en frecuencia del sistema en lazo cerrado cuando se utiliza el estimador sin deriva (6.6), y también se parametrizó la frecuencia de corte del filtro mediante la constante  $\varrho_c$  (comparar las ecuaciones (6.20) con (5.18), y los diagramas de Bode de las figuras 6.3 y 5.1).

Para el seguimiento de una trayectoria circular en el plano  $(x, y)$ , en la sección 6.5 se sintetizó una trayectoria de estado óptima (ver (6.27)) y se particionó la trayectoria circular en un conjunto finito de puntos estacionarios locales (ver (6.35) y (6.36)).

En la sección 6.6 se presentaron resultados experimentales cuando el cuadricóptero sigue una trayectoria circular de radio  $r_* = 5$  m en el plano  $(x, y)$ . Se consideraron dos casos: la trayectoria lenta que se recorre en 60 s y la trayectoria rápida que se recorre en 20 s. De estos resultados se obtuvo una reducción en los errores de seguimiento y guiñada al aplicar los estimadores sin deriva (6.22). En la trayectoria lenta se obtuvo una mejora en la eficiencia de alrededor del 10 % y en la trayectoria rápida una mejora en la eficiencia de alrededor del 20 %, y el error en guiñada se disminuyó en un factor de 1/3 (ver figuras 6.6 y 6.8).

## 7.1 Lista de publicaciones

### Revistas internacionales

- Moisés Bonilla, L.A. Blas, Vadim Azhmyakov, Michel Malabre y Sergio Salazar. “Robust structural feedback linearization based on the nonlinearities rejection”. Journal of the Franklin Institute, Elsevier, 2019. **En publicación**  
<https://doi.org/10.1016/j.jfranklin.2019.11.044>
- L.A. Blas, Moisés Bonilla, Sergio Salazar, Michel Malabre y Vadim Azhmyakov. “Synthesis of a robust structural feedback linearization scheme for an experimental quadrotor”. Control Engineering Practice, 2019. **Sometido**

### Congresos internacionales

- Moisés Bonilla, L.A. Blas, Sergio Salazar, J.C. Martinez y Michel Malabre. “A robust linear control methodology based on fictitious failure rejection”. 2016 European Control Conference (ECC), pp. 2596–2601
- L.A. Blas, Moisés Bonilla, Michel Malabre, Vadim Azhmyakov y Sergio Salazar. “Structural feedback linearization based on nonlinearities rejection”. IFAC, 50(1):945-950, 2017
- L.A. Blas, Moisés Bonilla, Sergio Salazar, Michel Malabre y Vadim Azhmyakov. “Synthesis of a robust linear structural feedback linearization scheme for an experimental quadrotor”. 2019 European Control Conference (ECC), pp. 1431–1436
- L.A. Blas, Moisés Bonilla, Sergio Salazar, Michel Malabre, Vadim Azhmyakov. “Robust tracking scheme for an experimental quadrotor”. IFAC, 2020 **Sometido**

# Apéndice A

## Cálculo de las constantes $k_F$ y $k_T$

Cada rotor tiene una velocidad angular  $\omega_i$  y produce una fuerza vertical  $F_i$  y un momento  $M_i$  que pueden medirse de manera experimental. A los datos experimentales se les aplica un ajuste de curva polinomial y las expresiones que resultan son las siguientes

$$F = k_F \omega^2, \quad (\text{A.1})$$

$$M = k_T \omega^2. \quad (\text{A.2})$$

En nuestro caso, el procedimiento para calcular la constante  $k_T$  se basa en emplear las expresiones para la potencia eléctrica, la potencia desarrollada por el par motor y la expresión (A.2).

La expresión para la potencia eléctrica esta dada por

$$P = IV, \quad (\text{A.3})$$

y la potencia desarrollada por el par motor es:

$$P = M\omega, \quad (\text{A.4})$$

donde  $P$  es la potencia en Watts,  $I$  es la corriente en Amperios,  $V$  el voltaje en voltios,  $M$  el par motor en Nm y  $\omega$  la velocidad angular en  $\frac{\text{rad}}{\text{s}}$ .

Sustituyendo la expresión (A.2) en (A.4) se obtiene

$$P = k_\tau \omega^3. \quad (\text{A.5})$$

Despejando  $k_\tau$  de (A.5) se tiene

$$k_\tau = \frac{P}{\omega^3}, \quad (\text{A.6})$$

donde las unidades son  $\frac{\text{Nm}}{\text{rpm}^2}$  y la potencia  $P$  se obtiene a partir de (A.3).

Para obtener la constante  $k_F$ , considere que en vuelo estacionario los rotores de cuadricóptero generan un empuje total  $F_T = mg$ . En este estado los empujes nominales de los rotores deben satisfacer

$$F_i = \frac{mg}{4}. \quad (\text{A.7})$$

Igualando la expresiones (A.1) y (A.7) y despejando  $k_F$  se obtiene

$$k_F = \frac{mg}{4\omega^2}. \quad (\text{A.8})$$

La potencia eléctrica medida total es de aproximadamente 155W y la potencia de cada motor es de aproximadamente 36.25 W. Las revoluciones promedio  $451.5 \frac{\text{rad}}{\text{s}}$ .

De los valores anteriores y de la expresiones (A.6) y (A.8) se tiene

		<b>Unidades</b>
$k_\tau$	$4.31 \times 10^{-9}$	Nm/rpm <sup>2</sup>
$k_F$	$1.98 \times 10^{-7}$	N/rpm <sup>2</sup>

Tabla A.1: Constantes  $k_F$  y  $k_\tau$ .

# Apéndice B

## Matriz de inercia del cuadricóptero

El momento de inercia es la resistencia que presenta un cuerpo a ser acelerado en rotación. Para un sistema de partículas y un eje arbitrario, el momento de inercia se define como la suma de los productos de las masas de las partículas por el cuadrado de la distancia  $r$  de cada partícula a dicho eje. Y para un cuerpo de masa continua se obtiene mediante una integral sobre el volumen del todo el cuerpo.

El cuadricóptero se puede considerar como un objeto rígido y para obtener la matriz de inercia, el cuadricóptero se puede dividir en cuerpos geométricos cuyos momentos de inercia respecto a un eje dado  $L_0$  se pueden determinar empleando el teorema de Steiner o de los ejes paralelos:

*El momento de inercia de un objeto con masa  $m$  respecto a un eje  $L_0$  está dado por:*

$$I_0 = I + d^2m \quad (\text{B.1})$$

*donde  $I$  es el momento de inercia del objeto respecto a un eje paralelo que pasa por su centro de masa y  $d$  es la distancia perpendicular entre los ejes [41].*

La suma de las matrices de inercia de cada una de las partes dará como resultado la matriz de inercia del objeto compuesto. En el cuadricóptero las partes que lo constituyen se pueden visualizar principalmente como cilindros y prismas rectangulares, ver figuras B.1 y B.2.



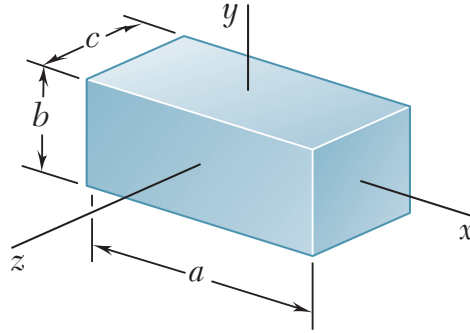


Figura B.1: Prisma.

Momentos de inercia para el prisma rectangular de la figura B.1.

$$I_x = \frac{1}{12}m(b^2 + c^2)$$

$$I_y = \frac{1}{12}m(c^2 + a^2) \tag{B.2}$$

$$I_z = \frac{1}{12}m(a^2 + b^2)$$

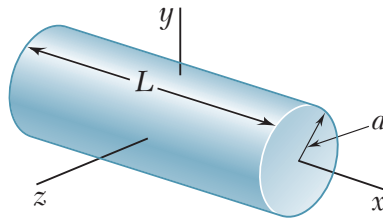


Figura B.2: Cilindro.

Momentos de inercia para el cilindro de la figura B.2.

$$I_x = \frac{1}{2}ma^2 \tag{B.3}$$

$$I_y = I_z = \frac{1}{12}m(3a^2 + L^2)$$

---

En la figura B.3 se muestra el cuadricóptero construido y al cual se le calculará la matriz de inercia.



Figura B.3: Plataforma experimental utilizada en esta tesis.

En la siguiente tabla se muestran los valores de la matriz de inercia obtenidos mediante fórmulas y el teorema de los ejes paralelos.

		<b>Unidades</b>
$I_{xx}$	0.0134	kg m <sup>2</sup>
$I_{yy}$	0.0140	kg m <sup>2</sup>
$I_{zz}$	0.0256	kg m <sup>2</sup>

Tabla B.1: Momentos de inercia.

El peso total del cuadricóptero es de 1.36 kg.



# Apéndice C

## Componentes del cuadricóptero

### C.1 Estructura mecánica del cuadricóptero

La estructura o armazón es el cuerpo principal del cuadricóptero. Todos los demás componentes (motores, baterías, piloto automático, etc.) son montados en la estructura.

La estructura del cuadricóptero es la turnigy Talon Quadcopter, esta presenta una estructura o marco principal y brazos de fibra de carbono, todos conectados entre si con piezas de aleación de aluminio. Construido con un peso ligero en fibra de carbono extremadamente rígida y la aleación de aluminio, el Talon ofrece una gran combinación de ahorro de peso y fuerza.

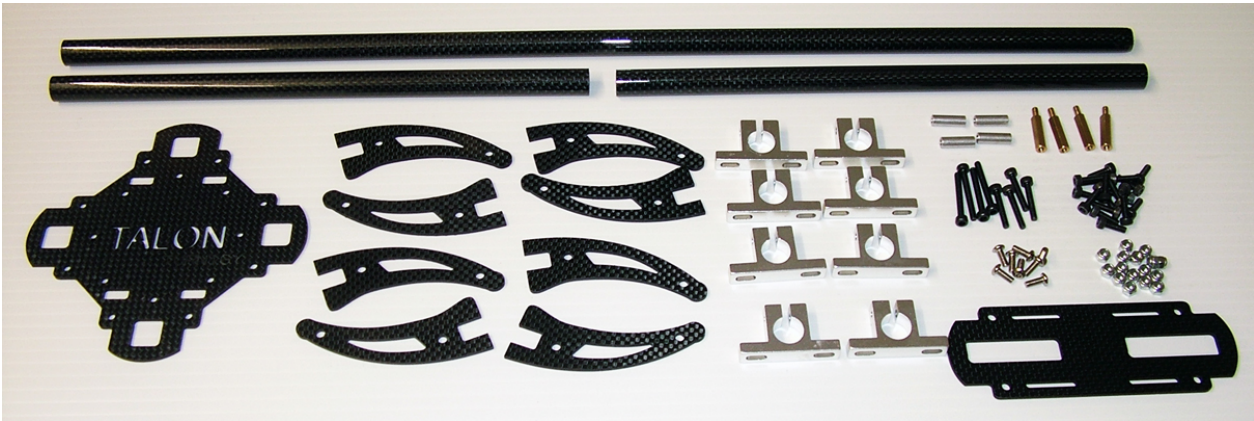


Figura C.1: Piezas de la estructura del cuadricóptero.

En la figura C.2 se muestra la estructura del cuadricoptero que se utilizó.



Figura C.2: Estructura del cuadricóptero.

Este armazón tiene las siguientes especificaciones y componentes

Peso	240g
Ancho	498 mm
Agujeros del perno de la montura del motor	22-32mm
Tubos de fibra de carbono	
Placa central de fibra de carbono	
Placa inferior de fibra de carbono	
Piernas de fibra de carbono	
Abrazaderas de aleación de aluminio	

Tabla C.1: Especificaciones y componentes.

## C.2 Sistema de propulsión

### C.2.1 Hélices

El fabricante del motor, que más adelante se describe, recomienda utilizar hélices 10x4.7 - 11x4.7. En nuestro caso, se utilizan las hélices Turnigy Slowfly 11x4.7, dos pares de hélices que giran en sentido horario (cw) y dos pares que giran en sentido antihorario (ccw).



Figura C.3: Hélices. (a) Hélice ccw, (b) hélice cw.

El número 11 corresponde a la longitud total de la hélice (largo de la hélice) y el segundo teóricamente es la distancia que recorre la hélice en una revolución, también denominado paso, ver figura C.4.

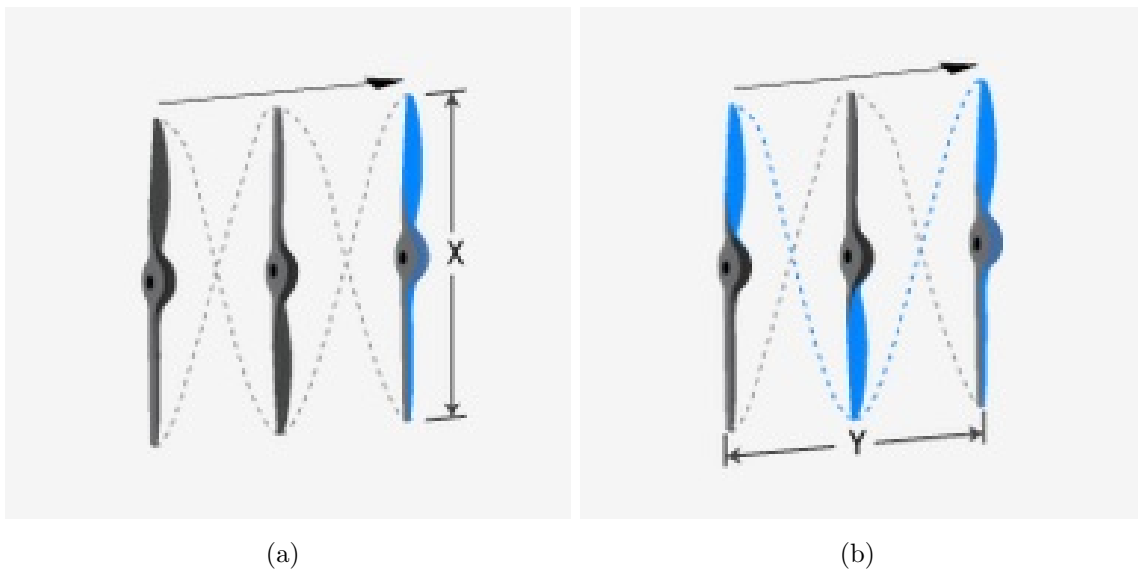


Figura C.4: Características de las hélices. (a) Diámetro de la hélice, (b) paso de la hélice.

### C.2.2 Motores brushless

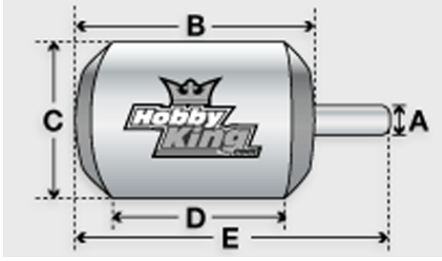
Se utilizan cuatro motores Turnigy Multistar 4220-880Kv 16Pole Multi-Rotor Outrunner. Estos cuentan con imanes 45SH de alta gama, rodamientos NMB de alta calidad, 12 polos, montaje de motor personalizado y todos están equilibrados con precisión para un funcionamiento suave. El 4220-880 incluye un adaptador de hélices que se ha construido en una montura de aluminio para una instalación rápida y fácil en la estructura de multi-rotor.



Figura C.5: Motor Turnigy Multistar 4220-880Kv 16Pole Multi-Rotor Outrunner.

Las especificaciones de los motores se muestran a continuación:





Constante de velocidad	880KV (rpm/V)
Celdas LiPo	3s
Potencia máxima	250 W
Corriente de trabajo	17 A
Corriente máxima (10s)	22 A
Corriente sin carga	0.8 A
Resistencia interna	0.066 $\Omega$
Número de polos	16
Diámetro del eje (A)	3 mm
Longitud (B)	21 mm
Diámetro (C)	42 mm
Longitud (D)	21 mm
Longitud total (E)	25 mm
Peso	65 g
Hélices	10x4.7 - 11x4.7

Tabla C.2: Características del motor.

### C.2.3 Controlador Electrónico de Velocidad (ESC)

Para la etapa de potencia y control de los motores brushless se utilizan cuatro controladores electrónicos de velocidad TURNIGY Plush 30amp Speed Controller w/BEC. Tienen una amplia gama de funciones de programación junto con una respuesta suave del acelerador.

## C.2. Sistema de propulsión

---

Programable, tanto vía controlador como con tarjeta de programación opcional. La tarjeta de programación es un elemento excelente, ya que instantáneamente le informa al usuario la configuración actual y con unos pocos clics de los botones, el usuario puede cambiar la configuración y tener la seguridad gráfica de ese cambio.



Figura C.6: Controlador de velocidad TURNIGY Plush 30amp Speed Controller w/BEC.

Las especificaciones de los controladores de velocidad son las siguientes:

Corriente constante	30 A
Corriente pico	40 A
Modo BEC	Lineal
BEC	5V / 2A
Baterías LiPo	2-4S
Peso	25 g
Tamaño	45x24x11mm

Tabla C.3: Características del controlador de velocidad.

Los ESC básicamente crean una salida de alimentación de AC trifásica de una fuente de alimentación de DC para hacer funcionar el motor mediante el envío de una secuencia de señales de AC generadas desde el circuito de los ESC.

## C.2.4 Batería

Para alimentar los componentes electrónicos del cuadricóptero, se utiliza una batería de polímero de litio Turnigy Graphene 4000mAh 3S 45C Lipo Pack w/XT90.

Los paquetes Turnigy Graphene utilizan carbono en la estructura de la batería para formar una sola capa de grafeno de solo 0.335 nm de grosor, lo que hace que ese tipo de sustrato de la batería sea el más delgado conocido por la humanidad. Las partículas de grafeno forman un compuesto muy denso que permite que los electrones fluyan con menos resistencia en comparación con las tecnologías tradicionales de baterías Lipoly. El resultado es una batería capaz de mantener una mayor potencia de salida mientras permanece mucho más fría bajo carga. Dado que el calor y la resistencia son el enemigo natural de las baterías, la química del grafeno ha reducido significativamente estos problemas y el resultado es un aumento increíble del ciclo de vida útil.



Figura C.7: Batería Turnigy Graphene 4000mAh 3S 45C Lipo pack w/XT90.

La batería tiene las siguientes especificaciones:

Capacidad (carga)	4000 mAh
Configuración de celdas	3S1P / 11.1 V
Constante de descarga	45C
Pico o ráfaga de descarga	90C
Peso	381 g
Dimensiones	144 × 51 × 27 mm
Conector de balanceo	JST-XH
Conector de descarga	XT-90

Tabla C.4: Características de la batería.

## C.3 Sistema de control y comando

### C.3.1 Transmisor DX7s

Tiene todas las funciones que un avión o piloto de helicóptero podría desear, pero no tienes que ser un experto para usarlas. La interfaz intuitiva de SimpleScroll<sup>TM</sup> hace que navegar por los menús y cambiar las configuraciones sea tan simple como “rodar y hacer clic”.



Figura C.8: Transmisor de 7 canales DX7s.

Número de canales	7
Modulación	DSM2/DSM
Banda	2.4 GHz
Receptor	AR8000
Funciones de programación	avión y helicóptero
Memoria	hasta 20 modelos
Modos	seleccionables 1, 2, 3 o 4
Tipo de batería del transmisor (Tx)	4 Celdas 2000mAh Ni-MH
Cargador	Adaptador internacional de AC

Tabla C.5: Especificaciones del Transmisor DX7s.

### C.3.2 Piloto automático Pixhawk

El piloto automático Pixhawk®1 es un popular controlador de vuelo de propósito general basado en el diseño de hardware abierto Pixhawk-project FMUv2 (combina la funcionalidad del PX4FMU + PX4IO), ver figura C.9.

Originalmente fabricado por 3DR®, esta placa era la plataforma de microcontrolador estándar original para PX4®. La tarjeta ya no es fabricada por 3DR.



Figura C.9: Piloto automático Pixhawk.

El Pixhawk tiene las siguientes especificaciones:

## Apéndice C. Componentes del cuadricóptero

Procesador	32-bit ARM Cortex M4 core con FPU 168 Mhz/256 KB RAM/2 MB Flash 32-bit failsafe co-processor
Sensores	ST Micro L3GD20H 16-bit gyroscope ST Micro LSM303D 14-bit acelerómetro/magnetómetro MEAS MS5611 barómetro MPU6000 acelerómetro/giroscopio
Potencia	Controlador ideal de diodos con conmutación automática Servo rail de alta potencia (7 V) y alta corriente lista Todas las salidas periféricas protegidas contra sobrecorriente, todas las entradas ESC protegidas
Interfaces	Puertos seriales 5x UART, 1 capacidad de alta potencia, 2x con control de flujo HW Entrada Spektrum DSM/DSM2/DSM-X Satellite Entrada y salida Futaba S.BUS Sumador de señal PPM Entrada RSSI (PWM o voltaje) I2C, SPI, 2x CAN, USB Entradas ADC 3.3 y 6.6
Dimensiones	Peso 38 g Anchura 50 mm Altura 15.5 mm Longitud 81.5 mm

Tabla C.6: Características del piloto automático Pixhawk.

El equipo completo del Pixhawk trae los siguientes componentes:

- a) Cable UBS

### C.3. Sistema de control y comando

---

- b) Módulo de potencia para la conexión de la batería LiPo
- c) Calcomanías guía de conexión “APM Rover”, “APM Copter” and “APM Plane”
- d) Interruptor de seguridad
- e) Cables conectores varios
- f) Módulo divisor I2C
- g) Adaptador USB de tarjeta micro SD
- h) Almohadillas de espuma para el amortiguamiento de vibraciones
- i) Adaptador y tarjeta micro SD de 4GB
- j) Zumbador

Estos se muestran en la figura C.10.



Figura C.10: Piloto automático Pixhawk y componentes.



## C.4 El módulo UBlox GPS + Compass

El módulo UBlox GPS + Compass es el GPS más utilizado para los controladores de vuelo compatibles con ArduPilot. Existen muchas versiones de estos módulos disponibles de varios fabricantes.

ArduPilot configura el GPS automáticamente después del inicio, por lo que no es necesario realizar ninguna calibración relacionada con el GPS.

El GPS mostrado en la figura C.11 incorpora la brújula digital HMC5883L, que proporciona un método conveniente para montar la brújula lejos de las fuentes de interferencia que puedan estar presentes en los límites del vehículo. Cuenta con una antena más grande y el chipset de próxima generación. Una excelente solución para multicopteros y rovers en particular, donde la precisión del GPS es primordial. Cuenta con circuitos activos para la antena de parche cerámico, batería de respaldo recargable para arranques en caliente y EEPROM I2C para la configuración de almacenamiento.



Figura C.11: GPS con módulo de brújula.

Módulo	ublox LEA-6H
Frecuencia de actualización	5Hz
Antena de parche cerámico	25 x 25 x 4 mm
LNA y filtro SAW	
Batería de respaldo de litio recargable	3V
Regulador de bajo ruido	3.3V
I2C EEPROM para almacenamiento de configuración	
Indicadores LED de alimentación y de posición	
Expuesto RX, TX, 5V y GND pad	
Dimensiones	38x38x8.5mm
Peso	16.8g

Tabla C.7: GPS con módulo de brújula.

## C.5 Otros componentes

### C.5.1 Arnés de distribución de energía

Para conectar los ESC al sistema de potencia del cuadricóptero se utiliza el Multistar XT60 de 4 x 3.5mm con arnés de distribución JST Plug Quadcopter.



Figura C.12: Arnés de distribución de energía.

## C.5.2 Módulo de alimentación

Muchos controladores de vuelo pueden comprarse con un módulo de alimentación analógico que proporciona una fuente de alimentación estable al controlador de vuelo y también admite la medición del voltaje de la batería y el consumo de corriente.

El módulo de alimentación proporciona suficiente potencia para el controlador de vuelo, el receptor y algunos periféricos de baja potencia (lidar, telemetría), pero no tiene suficiente potencia para servos o dispositivos de alta corriente

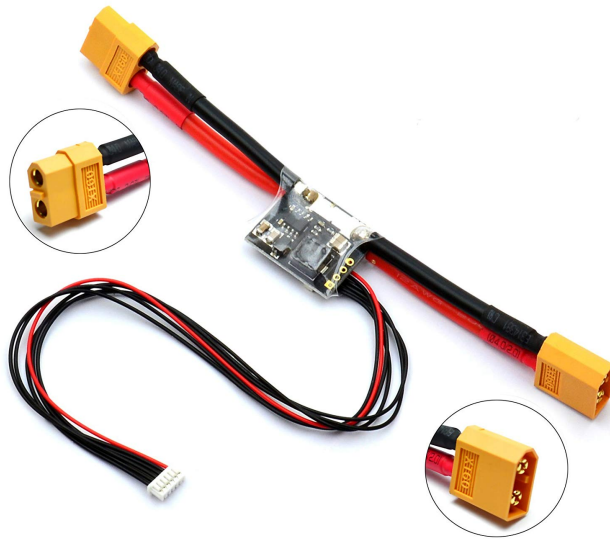


Figura C.13: Módulo de alimentación.

Voltaje de entrada máximo	18V (4S lipo)
Voltaje de entrada mínimo	4.5V
Corriente de sensado máxima	90A (pero solo es capaz de medir hasta 60A)
Configuración para medidas de voltaje y corriente	5V ADC
Salidas de regulador de conmutación	5.37V y 2.25A máximo

Tabla C.8: Módulo de alimentación.

### C.5.3 Cargador de baterías Lipo

El modelo B6AC V2 ahora ofrece conectividad a PC que le permite al usuario controlar el cargador a través de su ordenador personal y actualizar el firmware cuando sea necesario. Brinda la capacidad de ajustar el voltaje del terminal, lo que hace que el cargador sea la

solución de carga perfecta para baterías de servicio pesado. También se puede usar como medidor de batería de litio/resistencia interna y cuenta con funciones adicionales que incluyen el límite de corriente de carga automática, el límite de capacidad y el límite de tiempo de procesamiento, lo que convierte al B6AC V2 en uno de los cargadores más seguros disponibles en la actualidad.



Figura C.14: Cargador de baterías IMAX.

Voltaje de entrada AC	100 ~ 240V o entrada de 12V CC
Máxima carga	50W
Rango de corriente de carga	0.1 ~ 6.0A
Rango de corriente de descarga	0.1 ~ 1.0A
Celdas Ni-MH/NiCd	1 ~ 15
Celdas Li-ion/LiPoly	1 ~ 6
Pb voltaje de la batería	2 ~ 20v
Peso	632g
Dimensiones	135x144x36mm

Tabla C.9: Cargador de baterías.

# Apéndice D

## Código del cuadricóptero

### D.1 Instalación de las herramientas de compilación

Los pasos para instalar las herramientas para la compilación del código y generar el firmware personalizado que se carga al Pixhawk son los siguientes

1. Coloque la carpeta GitHub<sup>1</sup> en la carpeta Documentos. El contenido de la carpeta GitHub es el siguiente
  - ardupilot
  - PX4Firmware
  - PX4NuttX
  - uavcan
  - VRNuttX

2. Instale la cadena de herramientas

**px4\_toolchain\_installer\_v14\_win\_probado-rol.exe**

La cadena de herramientas PX4 incluye una versión preconfigurada de Eclipse que se ha configurado para el desarrollo de ArduPilot.

---

<sup>1</sup>El usuario puede nombrar esta carpeta como desee.

3. Instale la aplicación Mission Planner

**MissionPlanner-latest.msi**

Antes de continuar con el siguiente paso, es importante mencionar que no es necesario seguir el orden de instalación de los pasos antes descritos. A continuación se muestra cómo construir ArduPilot en Windows con Make.

4. Instrucciones de construcción para Pixhawk en Windows. Abra la consola PX4 y navegue hasta el directorio del vehículo de destino

- Inicia la consola PX4. Esto se puede encontrar en **Inicio - Aplicaciones - PX4 Toolchain - PX4 Console** (máquina con Windows 8.1) o puede ejecutar directamente `C:\px4\toolchain\msys\1.0\px4_console.bat`, esto desplegará la ventana mostrada en la figura D.1.



Figura D.1: Consola PX4.

- Vaya al directorio ArduPilot específico del vehículo en la consola PX4. En nuestro

## D.1. Instalación de las herramientas de compilación

---

caso en particular, para construir Copter navegue a

```
cd /c/USERS/ANGHEL/Documents/GitHub/ardupilot/ArduCopter
```

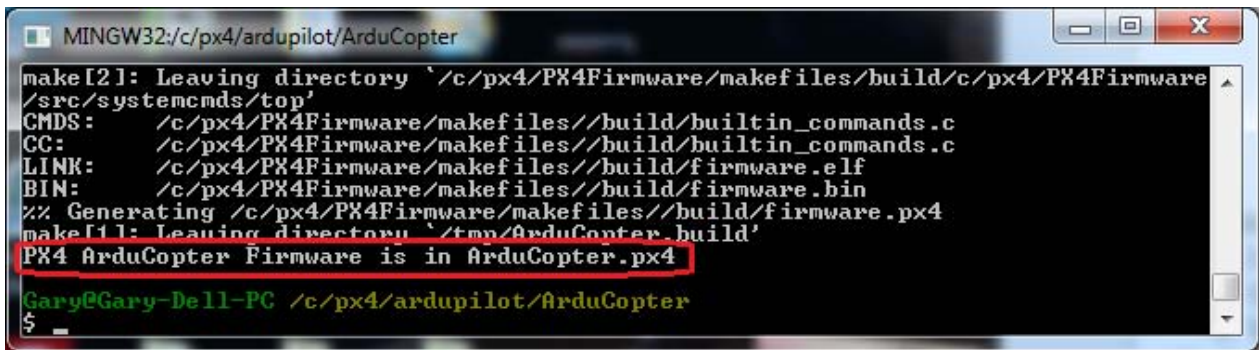
- Se escribe el siguiente comando para mostrar una lista del contenido del directorio (por defecto ordena la salida alfabéticamente)

```
ls
```

- Construye<sup>2</sup> el firmware Pixhawk para un cuadricóptero ingresando el siguiente comando

```
make px4-v2
```

El firmware se creará en el directorio ArduCopter con la extensión de archivo .px4, ver figura D.2.



```
MINGW32:/c/px4/ardupilot/ArduCopter
make[2]: Leaving directory `c:/px4/PX4Firmware/makefiles/build/c/px4/PX4Firmware
/src/systemcmds/top'
CMDS: /c/px4/PX4Firmware/makefiles//build/builtin_commands.c
CC: /c/px4/PX4Firmware/makefiles//build/builtin_commands.c
LINK: /c/px4/PX4Firmware/makefiles//build/firmware.elf
BIN: /c/px4/PX4Firmware/makefiles//build/firmware.bin
%% Generating /c/px4/PX4Firmware/makefiles//build/firmware.px4
make[1]: Leaving directory `tmp/ArduCopter_build'
PX4 ArduCopter Firmware is in ArduCopter.px4
Gary@Gary-Dell-PC /c/px4/ardupilot/ArduCopter
$
```

Figura D.2: Firmware creado en el directorio ArduCopter.

### 5. Configurar Eclipse para editar y crear binarios para las placas Pixhawk

#### (a) Iniciar Eclipse<sup>3</sup>

- En el menú Inicio de Windows (**Inicio - Aplicaciones - PX4 Toolchain - PX4 Eclipse**). Se obtiene la ventana mostrada en la figura D.3, o

---

<sup>2</sup>Esto puede tardar hasta 25 minutos.

<sup>3</sup>Asegúrate de tener instalado Java antes de que Eclipse pueda ejecutarse o de lo contrario obtendrás Java Error Code 13 debido a la falta de coincidencia de la versión de 32/64 bits.



- Puede ejecutar directamente el archivo `C:\Pixhawk_Toolchain\toolchain\msys\1.0\px4_eclipse.bat`. Se obtiene la ventana mostrada en la figura D.3.

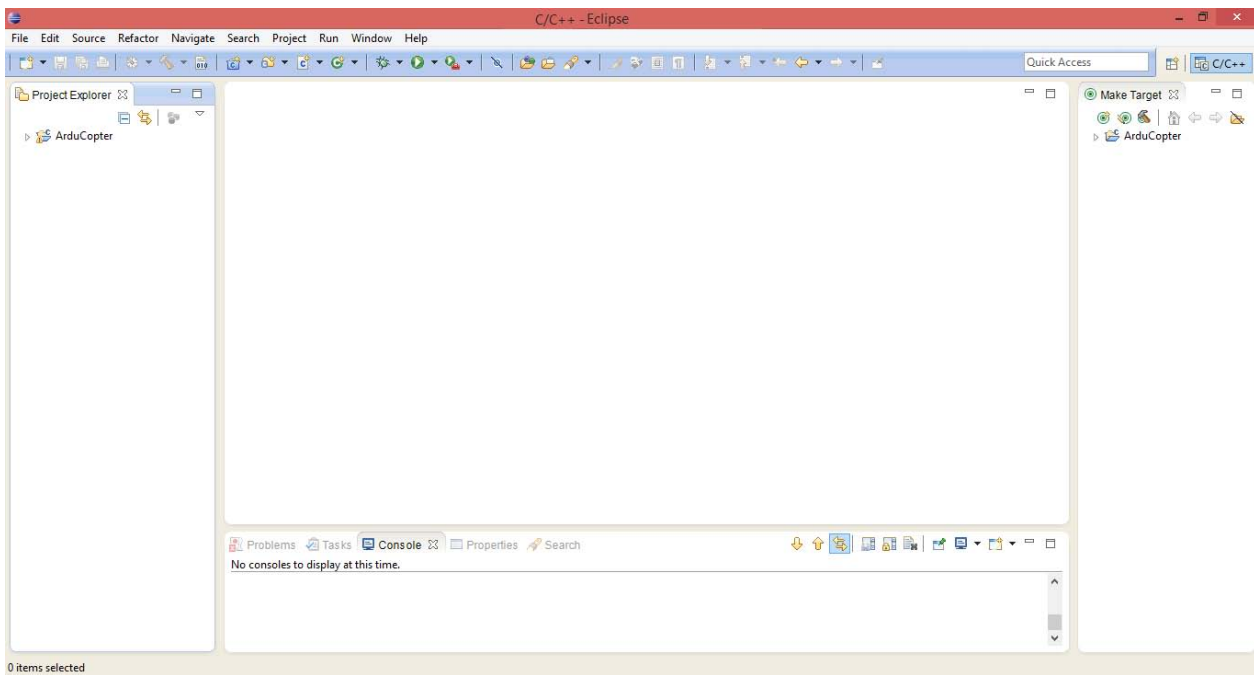


Figura D.3: Ventana de Eclipse.

- (b) Usa espacios en lugar de pestañas. Por defecto Copter, Plane y Rover usan espacios en lugar de pestañas. Esto se puede establecer en el valor predeterminado en Eclipse cambiando las siguientes dos configuraciones

- Seleccionar **Window - Preferences - General - Editors - Text Editors - Insert spaces for tabs**, ver figura D.4.

## D.1. Instalación de las herramientas de compilación

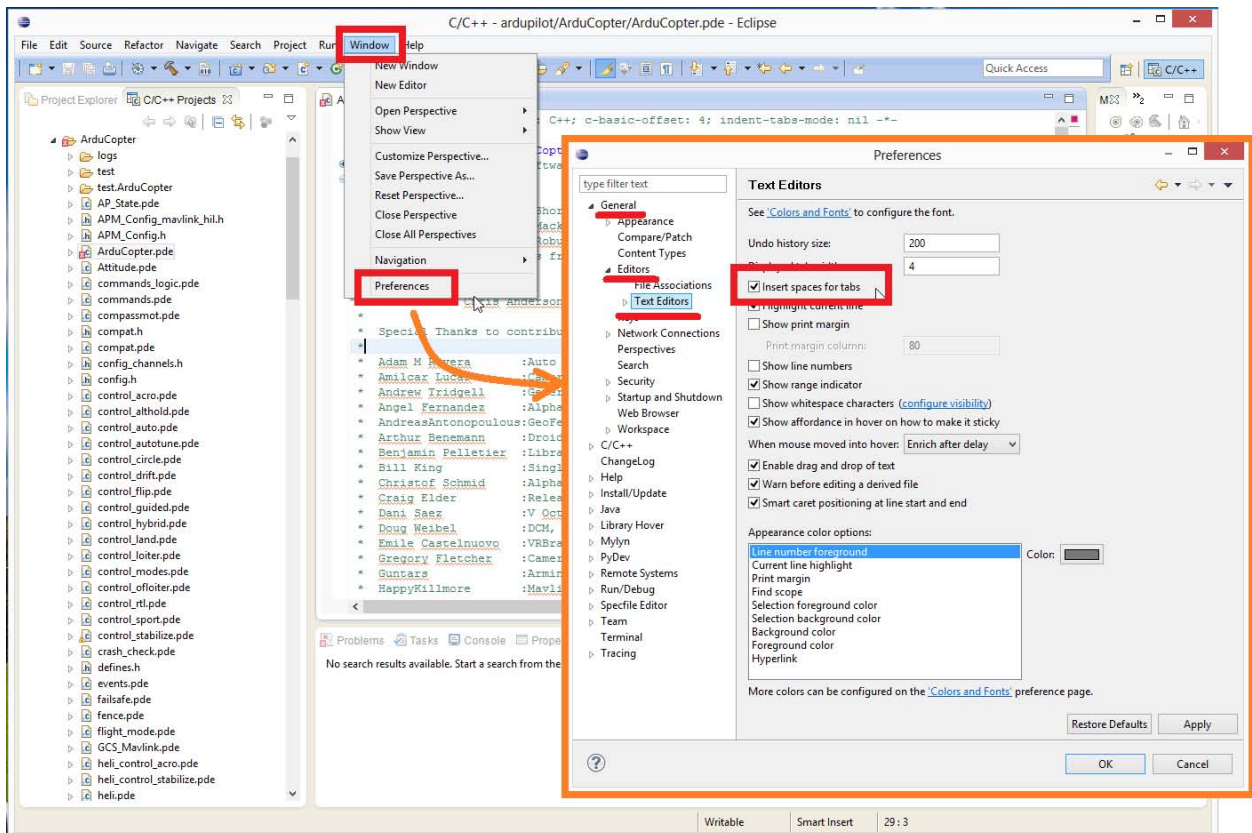


Figura D.4: Usar espacios en lugar de pestañas.

- Seleccionar **Windows - Preferences - C/C++ - Code Style - Formatter** y crea un nuevo perfil (Profile) (es decir, “K&R Tab”) que tiene “Indentation” (Sangría) establecida en “Spaces only” (Sólo espacios), ver figura D.5.

## Apéndice D. Código del cuadricóptero

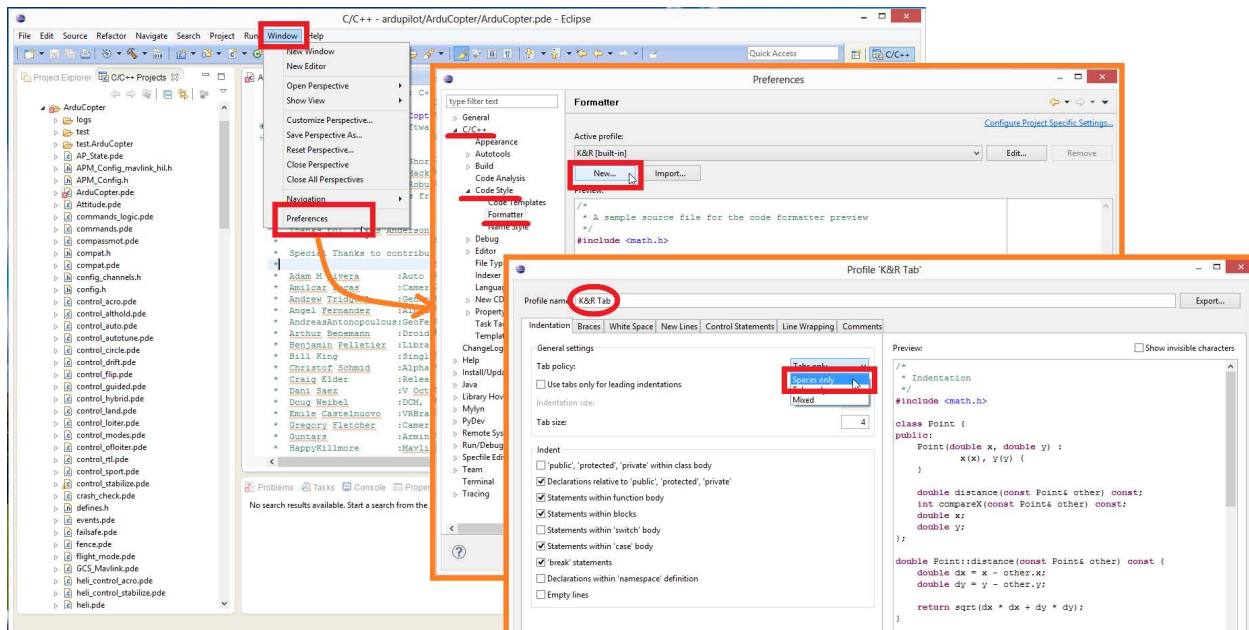


Figura D.5: “K&R Tab”.

- (c) Seleccionar **Windows - Preferences - C/C++ - File Types - New** para el tipo de archivo, en “modelo” (Pattern) escribir **\*.pde** y en “tipo” (Type) seleccionar **C++ Source File**, ver figura D.6.

## D.2. Mission Planner

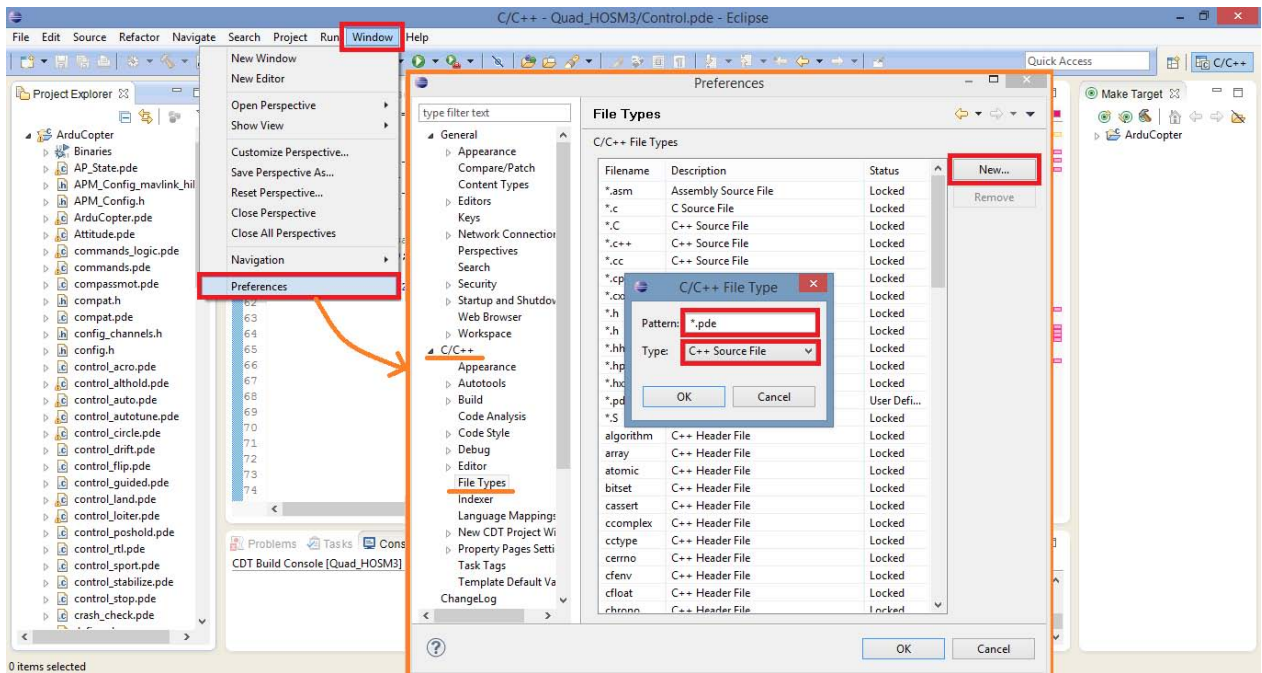


Figura D.6: Seleccionando el tipo de archivo.

Para mayor información sobre el paso 4, consulte el siguiente enlace

<http://ardupilot.org/dev/docs/building-px4-with-make.html>

Para el paso 5 consulte el siguiente enlace

<http://ardupilot.org/dev/docs/editing-the-code-with-eclipse.html>

## D.2 Mission Planner

Mission Planner fue creado por Michael Osborne, es una aplicación de estación de tierra con todas las funciones para el proyecto piloto automático de código abierto ArduPilot. Esta estación de control de tierra para avión, helicóptero y rover, es compatible sólo con Windows. Mission Planner se puede utilizar como una utilidad de configuración o como complemento de control dinámico de su vehículo autónomo. Estas son sólo algunas cosas que se pueden hacer con Mission Planner:

- Cargar el firmware (el software) en el piloto automático (Pixhawk) que controla el vehículo.
- Establecer, configurar y sintonizar el vehículo para un rendimiento óptimo.
- Planificar, guardar y cargar misiones autónomas en el piloto automático.
- Descargar y analizar los registros de la misión creados por su piloto automático.
- Interfaz con un simulador de vuelo PC para crear un simulador completo UAV hardware-in-the-loop.
- Con el hardware de telemetría adecuada puede:
  - Monitorear el estado del vehículo mientras está en funcionamiento.
  - Historial de los registros de telemetría que contienen mucha más información que los registros del piloto automático a bordo.
  - Ver y analizar los registros de telemetría.
  - Operar el vehículo en vista en primera persona (FPV por sus siglas en inglés).

En nuestro caso, Mission Planner lo usamos solamente para cargar el firmware personalizado al Pixhawk, en la figura D.7 se muestra la interfaz.

En el siguiente enlace

<http://ardupilot.org/planner/docs/mission-planner-installation.html>

se muestran los pasos a seguir para instalar Mission Planner, esa misma página también proporciona un enlace para descargar la aplicación más reciente de nombre **MissionPlanner-latest.msi**.

Otra opción para descargar la aplicación es ir al siguiente enlace

<https://github.com/ArduPilot/MissionPlanner>

para esto es necesario tener una cuenta en GitHub.

## D.3 Cargar el firmware personalizado al Pixhawk

1. Abrir Mission Planner **Inicio - Aplicaciones - Mission Planner**.



Figura D.7: Interfaz de inicio de la estación de control de tierra Mission Planner.

2. Para cargar el firmware personalizado al Pixhawk, en la pantalla de Mission Planner seleccione **Initial Setup - Install Firmware**, ver figura D.8.

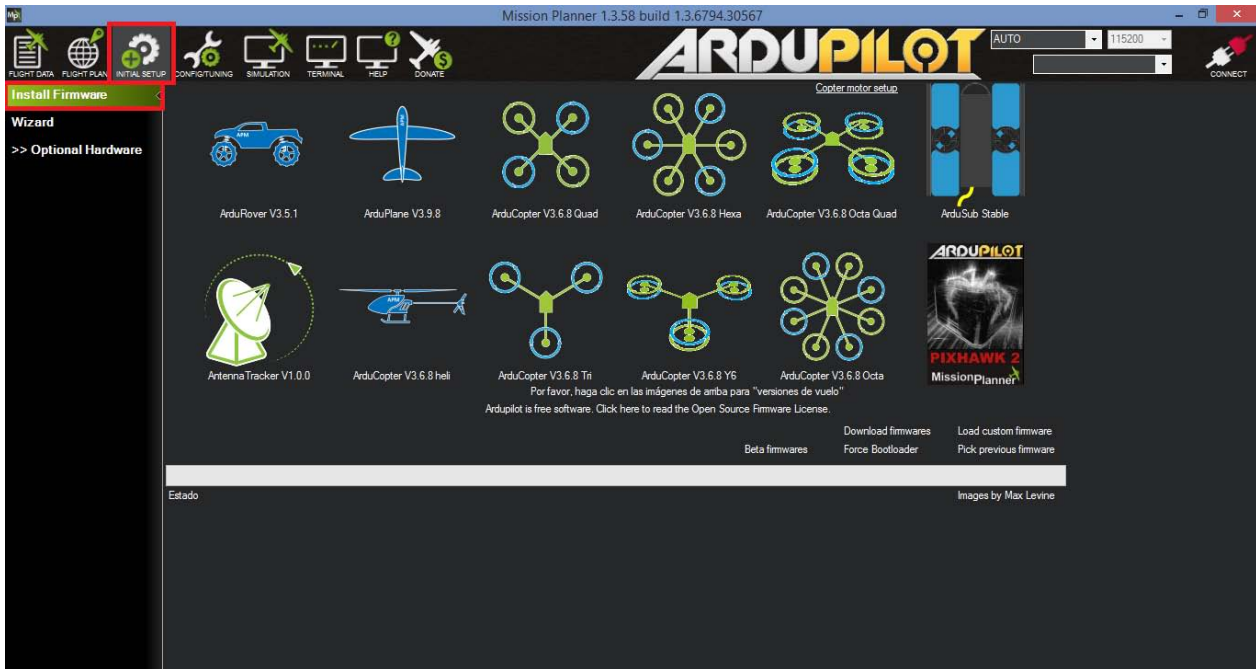


Figura D.8: Inicio de la instalación del firmware personalizado.

3. Conecte el Pixhawk usando el cable micro USB como se muestra en la figura D.9. Use un puerto USB directo en su computadora (no un concentrador USB). Si es la primera vez, Windows debería detectar e instalar automáticamente el software del controlador correcto.

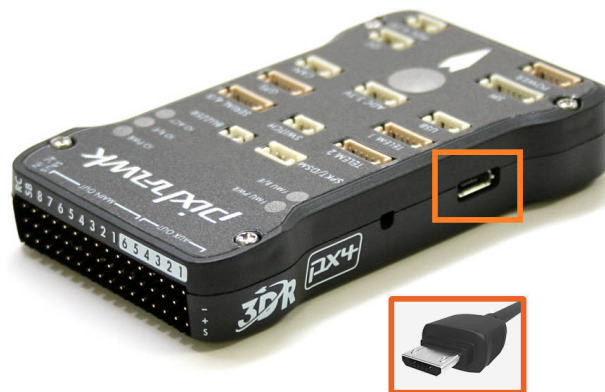


Figura D.9: Conexión USB del Pixhawk.

### D.3. Cargar el firmware personalizado al Pixhawk

4. En Mission Planner seleccione **Load custom firmware** (flecha roja con el número 1, ver figura D.10), esto desplegará una ventana en la que deberá buscar dentro de la carpeta **Documentos** (flecha roja con el número 2) la carpeta **GitHub** y dentro de esta última la carpeta **ardupilot**. Dentro de esta busque la carpeta con el nombre del proyecto que previamente ha trabajado y de su interior seleccione el archivo con extensión \*.px4<sup>4</sup>, y de clic en abrir (cuadro color naranja). Por ejemplo, la ruta para cargar el firmware **Quad\_Code\_BLAS8-v2.px4** es la siguiente  
**Documents\GitHub\ardupilot\Quad\_Code\_BLAS8\Quad\_Code\_BLAS8-v2.px4**.

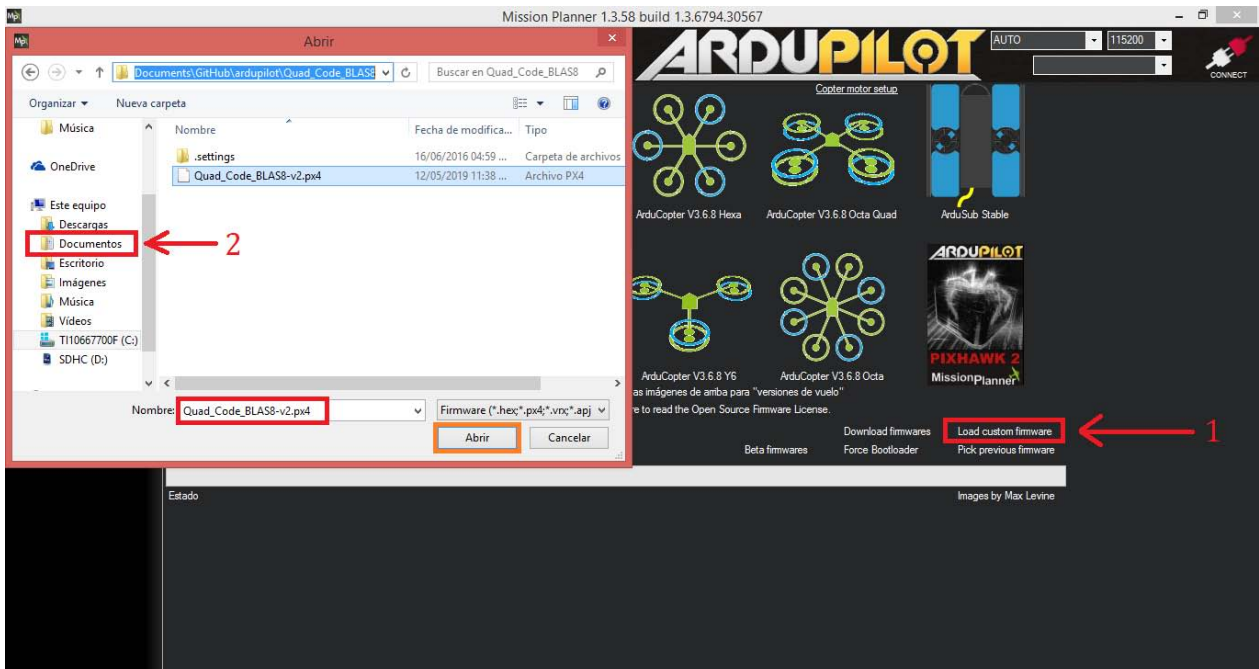


Figura D.10: Instalación del firmware personalizado.

5. Después de ejecutar el paso anterior, se mostrará la pantalla de instalación de firmware, ver figura D.11. La estación de tierra Mission Planner detectará la placa que se está utilizando (conectada vía USB), le pedirá que desenchufe la placa, presione OK y vuelva a enchufarla en unos pocos segundos (durante este breve período el cargador

<sup>4</sup>Es importante notar que la carpeta del proyecto y el archivo con extensión \*.px4 llevan el mismo nombre, además, el archivo \*.px4 después del nombre lleva -v2



de arranque acepta solicitudes para cargar el firmware, en nuestro caso el firmware personalizado).



Figura D.11: Instalar la solicitud de firmware.

- Si todo va bien, el progreso de carga del firmware se notará en el avance de la barra color verde, ver figura D.12. También verá que aparece un estado en la parte inferior izquierda, que incluye las palabras “erase...”, “program...”, “verify...” y “Upload Done”. El firmware se ha subido con éxito a la placa.

### D.3. Cargar el firmware personalizado al Pixhawk

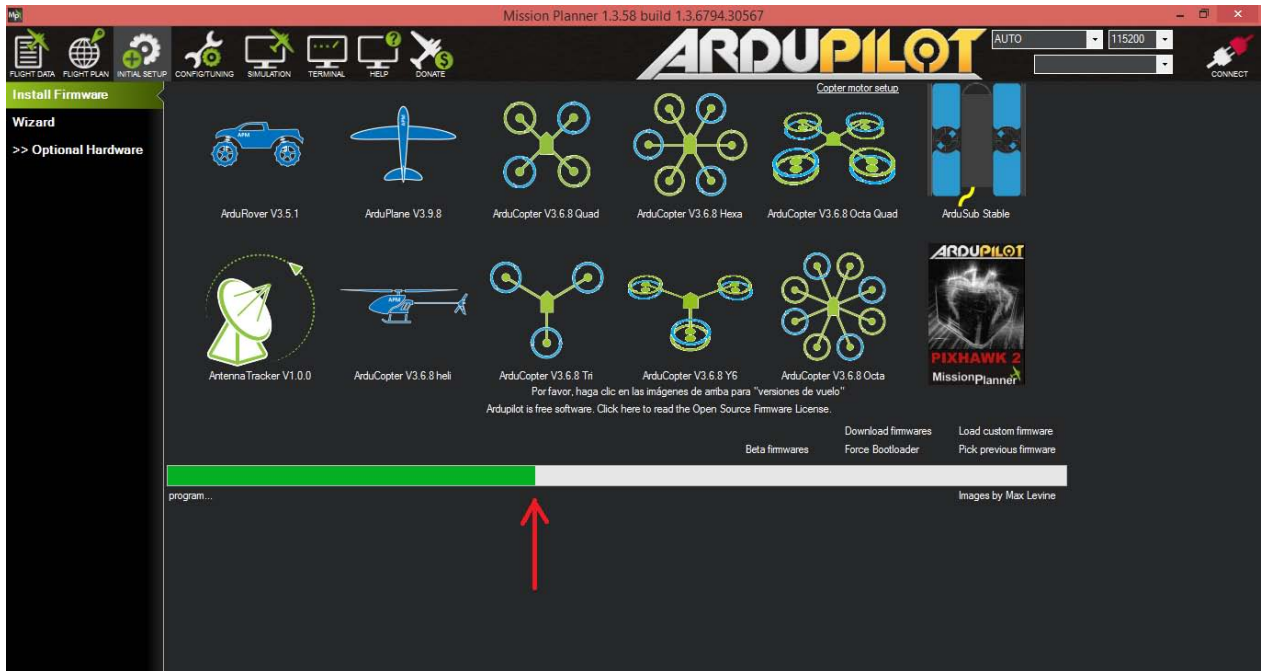


Figura D.12: Progreso de carga del firmware.

7. Se desplegará una ventana que indica esperar a que los tonos musicales terminen antes de hacer clic en OK, ver figura D.13.



Figura D.13: Firmware cargado en el Pixhawk.

Consultar el siguiente enlace

<http://ardupilot.org/planner/docs/common-loading-firmware-onto-pixhawk.html>

## D.4 Descripción del código personalizado

El código ArduCopter es el firmware desarrollado por la comunidad del proyecto ArduPilot que permite el vuelo de aeronaves. Esta característica la reúnen vehículos aéreos como los trirotos, cuadricópteros, hexaroteros, octaroteros o helicópteros.

El firmware completo se integra por archivos específicos del firmware ArduCopter y los archivos restantes componen la biblioteca común a todos los firmwares.

Para construir el código personalizado se ha utilizado como base el código ArduCopter versión V3.3-rc3. En el código personalizado casi la totalidad de las funciones aportadas por

## D.4. Descripción del código personalizado

---

el firmware completo no son utilizadas y solo se dejan las más indispensables.

En la figura D.14 se muestra de manera general la estructura que tiene el programa personalizado, el cual se compila para generar el firmware que posteriormente se carga en el piloto automático Pixhawk.

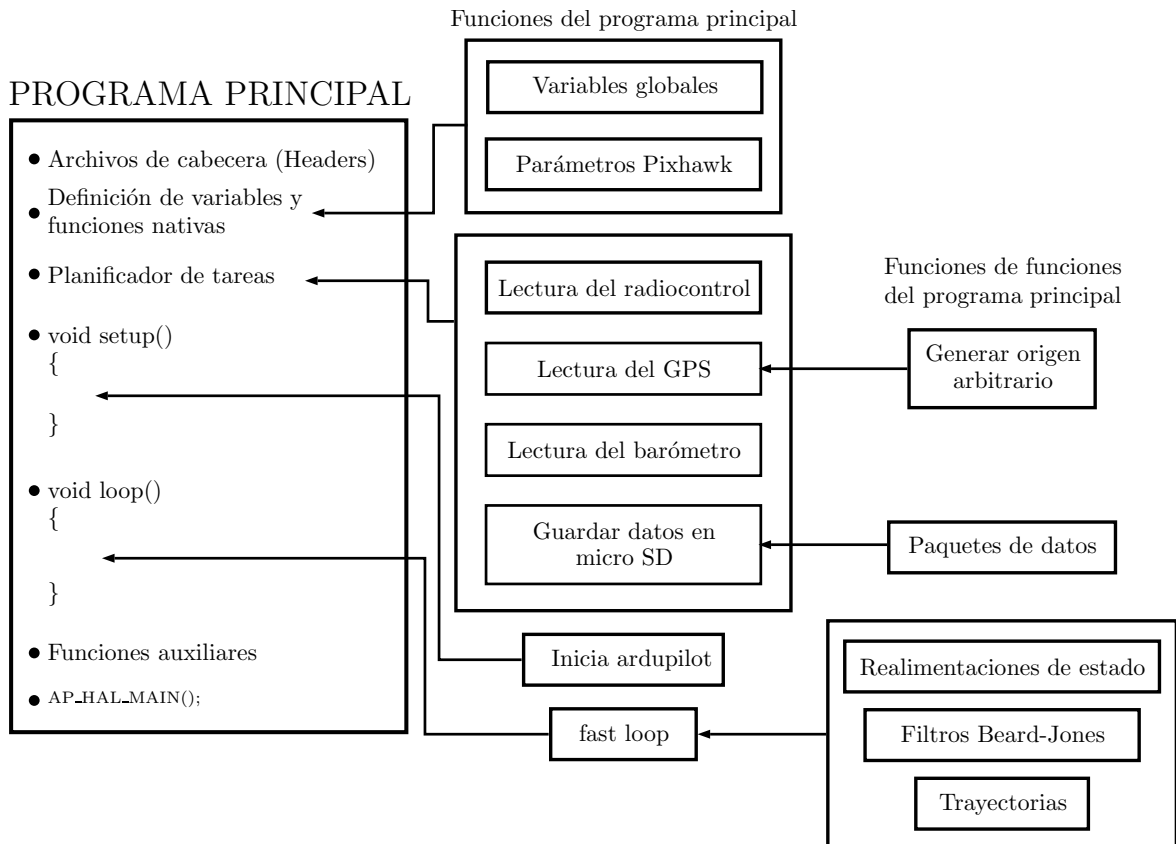


Figura D.14: Esquema general del programa personalizado.

A continuación se describen las líneas de código del programa principal.

## D.4.1 Programa principal

### D.4.1.1 Archivos de cabecera (Headers)

```
1
2 /*
3  * Quad_Code_BLAS8 Version 8.0
4  * Basado en código de Arducopter
5  * Creado por Anghel Blas
6  * DCA Cinvestav
7  */
8
9 #include <math.h>
10 #include <stdio.h>
11 #include <stdarg.h>
12 #include <stdlib.h>
13
14 // Common dependencies
15 #include <AP_Common.h>
16 #include <AP_Programm.h>
17 #include <AP_Menu.h>
18 #include <AP_Param.h>
19 #include <StorageManager.h>
20 // AP_HAL
21 #include <AP_HAL.h>
22 #include <AP_HAL_AVR.h>
23 #include <AP_HAL_SITL.h>
24 #include <AP_HAL_PX4.h>
25 #include <AP_HAL_VRBRAIN.h>
26 #include <AP_HAL_FLYMAPLE.h>
27 #include <AP_HAL_Linux.h>
28 #include <AP_HAL_Empty.h>
29
30 // Application dependencies
31 #include <GCS.h>
```

#### D.4. Descripción del código personalizado

---

```
32 #include <GCS_MAVLink.h>           // MAVLink GCS definitions
33 #include <AP_SerialManager.h>      // Serial manager library
34 #include <AP_GPS.h>                // ArduPilot GPS library
35 #include <DataFlash.h>             // ArduPilot Mega Flash Memory Library
36 #include <AP_ADC.h>                // ArduPilot Mega ADC Library
37 #include <AP_ADC_AnalogSource.h>
38 #include <AP_Baro.h>
39 #include <AP_Compass.h>            // ArduPilot Mega Magnetometer Library
40 #include <AP_Math.h>                // ArduPilot Mega Vector/Matrix math Library
41 #include <AP_Curve.h>              // Curve used to linearlise throttle pwm to thrust
42 #include <AP_InertialSensor.h>     // ArduPilot Mega Inertial Sensor Library
43 #include <AP_AHRS.h>
44 #include <AP_NavEKF.h>
45 #include <AP_Mission.h>            // Mission command library
46 #include <AP_Rally.h>              // Rally point library
47 #include <AC_PID.h>                // PID library
48 #include <AC_PI_2D.h>              // PID library (2-axis)
49 #include <AC_HELI_PID.h>           // Heli specific Rate PID library
50 #include <AC_P.h>                  // P library
51 #include <AC_AttitudeControl.h>    // Attitude control library
52 #include <AC_AttitudeControl_Heli.h> // Attitude control library for helicopter
53 #include <AC_PosControl.h>         // Position control library
54 #include <RC_Channel.h>            // RC Channel Library
55 #include <AP_Motors.h>              // AP Motors library
56 #include <AP_RangeFinder.h>        // Range finder library
57 #include <AP_OpticalFlow.h>        // Optical Flow library
58 #include <Filter.h>                // Filter library
59 #include <AP_Buffer.h>              // APM FIFO Buffer
60 #include <AP_Relay.h>              // APM relay
61 #include <AP_ServoRelayEvents.h>
62 #include <AP_Camera.h>             // Photo or video camera
63 #include <AP_Mount.h>              // Camera/Antenna mount
64 #include <AP_Airspeed.h>           // needed for AHRS build
65 #include <AP_Vehicle.h>            // needed for AHRS build
```

```
66 #include <AP_InertialNav.h> // ArduPilot Mega inertial navigation library
67 #include <AC_WPNav.h> // ArduCopter waypoint navigation library
68 #include <AC_Circle.h> // circle navigation library
69 #include <AP_Declination.h> // ArduPilot Mega Declination Helper Library
70 #include <AC_Fence.h> // Arducopter Fence library
71 #include <SITL.h> // software in the loop support
72 #include <AP_Scheduler.h> // main loop scheduler
73 #include <AP_RCMapper.h> // RC input mapping library
74 #include <AP_Notify.h> // Notify library
75 #include <AP_BattMonitor.h> // Battery monitor library
76 #include <AP_BoardConfig.h> // board configuration library
77 #include <AP_Frsky_Telem.h>
78 #if SPRAYER == ENABLED
79 #include <AC_Sprayer.h> // crop sprayer library
80 #endif
81 #if EPMENABLED == ENABLED
82 #include <AP_EPM.h> // EPM cargo gripper stuff
83 #endif
84 #if PARACHUTE == ENABLED
85 #include <AP_Parachute.h> // Parachute release library
86 #endif
87 #include <AP_LandingGear.h> // Landing Gear library
88 #include <AP_Terrain.h>
```

Las líneas 9 a 88 corresponden a los archivos de cabecera (header file), que contienen elementos que normalmente se consideran compartidos con otras partes del código, como prototipos de funciones, una declaración directa de clases, subrutinas, variables, declaración externa para variables globales u otros identificadores, etc.

### D.4.1.2 Definición de variables y funciones nativas

Aquí se definen las variables globales y las funciones nativas propias del código ArduPilot para una compilación correcta.

```
90 #include "parameters.h"
91 #include "defines.h"
```

En la línea 90 se definen los parámetros (variables globales) que se utilizan en el código, por ejemplo: las ganancias de las realimentación de estado, las matrices de los observadores, etc. En la línea 91 se definen parámetros para controlar el brillo del led del Pixhawk y parámetros de registro para el almacenamiento de datos en la memoria micro SD.

```
93 const AP_HAL::HAL& hal = AP_HAL_BOARD_DRIVER;
```

La instancia AP\_HAL, es la capa de abstracción de hardware para el proyecto ArduPilot. La clase AP\_HAL::HAL es un contenedor para un conjunto completo de controladores de dispositivo.

```
101 static AP_GPS gps;
102 static AP_Baro barometer;
103 static AP_InertialSensor ins;
104
105 static AP_SerialManager serial_manager;
106 static Compass compass;
107
108 #if CONFIG_SONAR == ENABLED
109 static RangeFinder sonar;
110 static bool sonar_enabled = true;
111 #endif
```

Declaración de objetos de clase que manejan los datos de los sensores. Las líneas 101 a 103 corresponde a tres objetos que manejan los principales sensores a bordo (gps, barómetro, acelerómetro, magnetómetro y giroscopios). Estos tres objetos son manejados por ArduPilot para calcular la posición del vehículo. En las líneas 108 a 111 se habilita el sonar.

```
114 #if AP_AHRS_NAVEKF_AVAILABLE
115 AP_AHRS_NavEKF ahrs(ins, barometer, gps, sonar);
116 #else
```



```
117 AP_AHRS_DCM ahrs(ins, barometer, gps);  
118 #endif
```

Se utiliza un algoritmo de filtro de Kalman extendido (EKF) para estimar la posición del vehículo, la velocidad y la orientación angular en función de las medidas de los giroscopios, acelerómetros, brújula, GPS, velocidad del aire y presión barométrica. La fusión de todas las mediciones disponibles es mejor para rechazar las mediciones con errores significativos. Esto hace que el vehículo sea menos susceptible a las fallas que afectan a un solo sensor.

En las líneas 115 y 117 se muestra que el objeto `ahrs` es creado de la combinación de cuatro sensores (`gps`, `barómetro`, `ins`, `sonar`) y tres sensores (`gps`, `barómetro`, `ins`). AHRS (Attitude and Heading Reference System) es como un sensor inercial, proporciona orientación en 3D. De la biblioteca AHRS de ArduCopter, se usa la matriz de cosenos directores (DCM<sup>5</sup> por sus siglas en inglés) para calcular la solución de orientación.

```
121 static AP_InertialNav_NavEKF inertial_nav(ahrs);
```

En la línea 121 objeto `inertial_nav` da toda la información a cerca de nuestro vehículo como posición y velocidad. Combina los datos del acelerómetro con los del `gps` y los datos del barómetro para mejorar la altitud y la posición. Se usan los algoritmos de filtro complementarios.

```
126 static Vector3f pos, vel;  
127 bool entra_home = true;  
128 uint8_t flag_gps = 0;  
129 static uint32_t last_update;  
130 bool entra_offset = true;  
131 static Vector3f pos_offset, pos_offset1, vel_offset;  
132 static Vector3f pos_xyz, vel_xyz, pos_xyz1, vel_xyz1;
```

En las líneas 126 a 132 se definen las variables para la posición y velocidad del cuadricóptero. En la línea 129, `Vector3f` es para definir un vector de tres elementos en el que se almacenan datos de punto flotante. La línea 130 define una variable booleana la cual puede

---

<sup>5</sup>Es un algoritmo que es un equivalente de procesamiento menos intensivo del filtro de Kalman.

#### D.4. Descripción del código personalizado

---

ser verdadera o falsa. En las líneas 131 y 132 se definen las variables para generar el origen arbitrario en posición  $(x, y)$ , se utiliza en las realimentaciones de estado cuando se realiza el vuelo estacionario. En estas líneas también se definen las variables para las velocidades lineales.

```
137 Vector3f Gyro;
```

En la línea 137 se define el vector Gyro para asignarle las tres velocidades angulares.

```
142 float roll , Droll , pitch , Dpitch , yaw , Dyaw;
```

```
143 float Ctrl_Roll , Ctrl_Pitch , Ctrl_Yaw;
```

```
144 float pitchT , DpitchT;
```

```
145 float rollT , DrollT;
```

En las líneas 142 a 145 se definen las variables para el control por realimentación de estado en orientación (roll, pitch, yaw).

```
150 float z_ref_F;
```

```
151 float Ctrl_z;
```

```
152 bool Adq_z_ref = true;
```

En las líneas 150 a 152 se definen las variables para adquirir la altura deseada y para el control por realimentación de estado en altura.

```
157 float px , Dx , py , Dy;
```

```
158 float Ctrl_x , Ctrl_y;
```

En las líneas 157 a 158 se definen las variables para el control por realimentación de estado en posición  $(x, y)$ .

```
163 float m1_Ctrl , m2_Ctrl , m3_Ctrl , m4_Ctrl , m1_rc , m2_rc , m3_rc , m4_rc;
```

```
164 float m1 , m2 , m3 , m4;
```

```
165 float m1_obs , m2_obs , m3_obs , m4_obs;
```

Las líneas 163 a 165 corresponden a las variables que se utilizan para las señales que se envían a los motores, estas se obtienen del radiocontrol, controles de realimentación en orientación y posición, observadores, etc.

```
170 static ToshibaLED_PX4 toshiba_led;
```

En la línea 170 se declara el objeto de clase `toshiba_led` para el led del Pixhawk.

```
175 static uint32_t count0 = 0, count1 = 0, count2 = 0, ct3 = 0;
```

```
176 float t_t1 = 0, t_t2 = 0, tt7 = 0, fg = 0;
```

En las líneas 175 a 176 se definen las variables que se utilizan en los contadores y temporizadores para las trayectorias.

```
181 #if CONFIG_HAL_BOARD == HAL_BOARD_PX4
```

```
182 static DataFlash_File DataFlash("/fs/microsd/APM/LOGS");
```

```
183 #endif
```

En las líneas 181 a 183 se hace la configuración para guardar archivos de datos de vuelo en la memoria micro SD.

### D.4.1.3 Planificador de tareas

Debido a que los sensores trabajan a diferentes frecuencias, es necesario usar un planificador de tareas. En este se define la tarea a ejecutar, seguida de un primer número que corresponde a la frecuencia con la que debería llamarse cada tarea (en unidades de 2.5 ms) y el segundo número es el tiempo máximo que se espera que tome ejecutarse cada tarea (en microsegundos).

```
188 static AP_Scheduler scheduler;
```

```
189 static const AP_Scheduler::Task scheduler_tasks [] PROGMEM = {
```

```
190     { Read_radio,          4,    10 },
```

```
191     { GPS_call,           8,    90 },
```

```
192     { Barometro_call,     40,   70 },
```

```
193     { SaveMemorySD_call,  20,   50 },
```

```
194 };
```

En la línea 188 se declara el objeto `scheduler` para la planificación de tareas. En las líneas 189 a 194 definimos las tareas que se van a ejecutar: radiocontrol a 100 Hz, GPS a 50 Hz,

barómetro a 10 Hz y el almacenamiento de datos en la memoria micro SD a 20 Hz. Cada una de estas tareas se describen más adelante.

### D.4.1.4 void setup()

Dentro de la función void setup colocamos aquello que debemos ejecutar solo una vez, esto es, la inicialización de sensores, escritura a motores, etc.

```
196 void setup()
197 {
198     ins.init(AP_InertialSensor::COLD_START, AP_InertialSensor::RATE_400HZ);
199     init_ardupilot();
200     scheduler.init(&scheduler_tasks[0],
201                  sizeof(scheduler_tasks)/sizeof(scheduler_tasks[0]));
202     hal.scheduler->delay(1000);
203 }
```

En la línea 198 se inicializa la central inercial, en la línea 199 se llama la función **init\_ardupilot** que se describe en D.4.2.7. En esta función se inicializan sensores, se habilitan canales de salida para la escritura a motores, etc. En las líneas 200 y 201 inicializa el planificador de tareas. Y en la línea 202 se introduce un retardo antes de entrar al void loop.

### D.4.1.5 void loop()

La función loop se repite de forma indefinida hasta que se apague el Pixhawk y se ejecuta a una frecuencia de 400Hz. Es aquí donde se escribe el código para el control del cuadricóptero.

```
204 void loop()
205 {
206     ins.wait_for_sample();
207     uint32_t timer = hal.scheduler->micros();
208     fast_loop();
```

```
209     scheduler . tick ( ) ;
210
211     uint32_t time_available=(timer+MAIN_LOOP_MICROS)-hal.scheduler->micros ( ) ;
212     scheduler . run ( time_available ) ;
213 }
```

La línea 206 indica la espera de una nueva medición. En la línea 207 se adquiere el tiempo en milisegundos, este se utiliza en el planificador de tareas. La línea 208 llama a la función **fast\_loop** que se describe en D.4.2.8. La línea 209 le indica al planificador de tareas una unidad de tiempo en cada ciclo. La línea 211 calcula el tiempo disponible en el planificador de tareas. La línea 212 aplica al planificador de tareas el tiempo disponible calculado.

### D.4.1.6 Funciones auxiliares

```
310 float sat_signal(float dato, float upper_value, float lower_value)
311 {
312     if(dato >= upper_value) dato = upper_value;
313     if(dato <= lower_value) dato = lower_value;
314     return dato;
315 }
```

Las líneas de código 310 a 315 permiten obtener una señal de salida que es el valor de la señal de entrada limitada a los valores de saturación superior e inferior. Esta función se utiliza en las señales de control.

### D.4.1.7 AP\_HAL\_MAIN();

El código principal del cuadricóptero siempre se debe finalizar con la línea de código que se muestra en 317.

```
317 AP_HAL_MAIN ( ) ;
```

## D.4.2 Funciones del programa principal

### D.4.2.1 Variables globales

Las estructuras permite almacenar un conjunto de datos de diferente tipo, en el siguiente código se utilizan varias de estas estructuras para almacenar la mayoría de los datos que se usan en el programa principal y en las funciones (radiocontrol, control por realimentación, estimadores, generación de trayectorias).

La forma de utilizar los campos contenidos en la estructura se realiza de la siguiente manera, por ejemplo: para utilizar la ganancia  $K_p$  en alabeo (roll) se escribe `PDroll.Kp`, donde `PDroll` es el nombre de la variable y  $K_p$  es el nombre del campo, lo anterior corresponde al valor 172 (ver líneas 22 y 25).

```
1 /*
2  *      Author: ANGHEL
3  */
4
5 #ifndef PARAMETERS_H
6 #define PARAMETERS_H
7
8 // Variables para los canales del radiocontrol
9 struct radio_control{
10     int rc_Roll;
11     int rc_Pitch;
12     int rc_Yaw;
13     int rc_Throttle;
14     int Aux_1;
15     int Aux_2;
16     int Aux_3;
17 }rc;
18
19
20 // Ganancias para las realimentaciones en orientacion
21 struct Gain_PDs_Attitude{
```

```

22     float Kp;
23     float Kd;
24 }
25 PDroll = {172,45},
26 PDpitch = {172,45},
27 PDyaw = {110,85};
28
29
30 // Ganancias para las realimentaciones en (x,y,z)
31 struct Gain_PDs_Position{
32     float Kp;
33     float Kd;
34 }
35 PDx = {-30,-32.4},
36 PDy = {30,32.4},
37 PDz1 = {140,70},
38 PDz2 = {102.95,51.471};
39
40
41 // Variables para el barómetro
42 struct Baro{
43     float z_baro;
44 }baro={0};
45
46
47 // m y g
48 struct mg{
49     float m;
50     float g;
51 }param={1.36,9.81};
52
53
54 // Datos para el estimador sin deriva del subsistema (x-\theta)
55 //a=10, eps = 1/50

```

#### D.4. Descripción del código personalizado

---

```
56 struct MObs1_xt{
57     float Aex[4][4] = {{0.7975,0,0,0},{0,0.8012,0,0},
58                       {0,0,0.8504,0},{0,0,0,0.9996}};
59     float Bex[4] = {-1556.3647,1612.6719,-63.8010,-0.0001163};
60     float Crx[4] = {1,1,1,1};
61     float Drx = -1;
62     float GfKlx = -4129.2165;
63     float u_x = 0.0;
64     float vbcix[4] = {1.1435,-1.2539,0.1182,-0.0078};
65     float Out_xx_k_1[4] = {0,0,0,0};
66     float Out_xy_k = 0.0;
67     float out_xx_k_1[4] = {0,0,0,0};
68     float out_xy_k = 0.0;
69     float rhoX = 0.0110906;
70 }Obsxt1;
71
72
73 // Datos para el estimador sin deriva del subsistema (x-\phi)
74 //a=10, eps = 1/50
75 struct MObs1_yp{
76     float Aey[4][4] = {{0.7975,0,0,0},{0,0.8012,0,0},
77                       {0,0,0.8504,0},{0,0,0,0.9996}};
78     float Bey[4] = {-1627.8854,1686.7437,-66.7119,-0.0001215};
79     float Cry[4] = {1,1,1,1};
80     float Dry = -1;
81     float GfKly = 3952.25;
82     float u_y = 0.0;
83     float vbciy[4] = {-1.0075,1.1047,-0.1041,0.0069};
84     float Out_yy_k_1[4] = {0,0,0,0};
85     float Out_yy_k = 0.0;
86     float out_yy_k_1[4] = {0,0,0,0};
87     float out_yy_k = 0.0;
88     float rhoY = 0.0114568;
89 }Obsyp1;
```



```

90
91
92 // Datos para el estimador sin deriva del subsistema (\psi)
93 //a=40, eps = 1/10
94 struct MObs1_p{
95     float sysdRpA [2][2] = {{0.58936434,0},{0,0.99673477}};
96     float sysdRpB [2] = {-6.2023254,0.00069678556};
97     float sysdRpC [2] = {0.79468217,0.99836738};
98     float sysdRpD = -4.1008143;
99     float GfKlp = 62.30016;
100    float u_p = 0.0;
101    float vbcip [2] = {-1.3269236,1.4141901};
102    float Out_pp_k_1 [2] = {0,0};
103    float Out_py_k = 0.0;
104    float out_pp_k_1 [2] = {0,0};
105    float out_py_k = 0.0;
106    float rhoP = 0.27832;
107 }Obsp1;
108
109
110 // Datos para el seguimiento de trayectoria del subsistema (x-\theta)
111 struct vect_Rx{
112     float t_hover = 1200;
113     float alpha = 5;
114     float rr = 5;
115     float xxf = 10;
116     float Np = 36;
117     float Ttr = 1;
118     float vect_x0 [4] = {0,0,0,0};
119     float vect_x1 [4] = {0,0,0,0};
120     float MExpXTtr [4][4] = {{0.61183645,0.45807925,-1.1060239,-0.0003436615},
121                               {-0.7364176,-0.1841432,-0.2736955,-0.0000853243},
122                               {0.01863791,0.09521341,-0.2909483,-0.0000904268},
123                               {-0.19377184,-0.1908063,0.1763713,0.0000546899}};

```

#### D.4. Descripción del código personalizado

---

```
124 float vec_x0[4] = {0,0,0,0};
125 float xbx[4] = {0,0,0,0};
126 float Ubxt[4] = {0,0,0,0};
127 float ux = 0.0;
128 float x_k[4] = {0,0.87266463,0,0.0027097742};
129 float mAx[4][4] = {{0.99999998,0.002499978,-0.000030539316,0},
130                    {-0.000016286635,0.99998238,-0.024431453,0},
131                    {0.001328166,0.0014372494,0.99237128,0.00049584863},
132                    {1.0625328,1.1497995,-6.1029735,-0.60332109}};
133 float mBx[4] = {0,0,0.0000442722,0.03541776};
134 float mCx[4][4] = {{0.99999999,0.001249989,-0.000015269658,0},
135                    {0,0.99999119,-0.012215726,0},
136                    {0.00066408299,0.00071862469,0.99618564,0.0002479243},
137                    {0.53126639,0.57489975,-3.0514867,0.19833945}};
138 float mDx[4] = {0,0,0.0000221361,0.01770888};
139 float Out_Xx_k-1[4] = {0,0,0,0};
140 float Out_Xy_k[4] = {0,0,0,0};
141 float out_Xx_k-1[4] = {0,0,0,0};
142 float out_Xy_k[4] = {0,0,0,0};
143 }vRx;
144
145
146 // Datos para el seguimiento de trayectoria del subsistema (y-\phi)
147 struct vect_Ry{
148     float alpha = 5;
149     float rr = 5;
150     float yyf = 10;
151     float Np = 36;
152     float Ttr = 1;
153     float vect_y0[4] = {0,0,0,0};
154     float vect_y1[4] = {0,0,0,0};
155     float MExpyTtr[4][4] = {{0.61182663,0.45806864,1.1059676,0.00032893429},
156                             {-0.73642005,-0.1841458,0.2736815,0.00008165628},
157                             {-0.01863533,-0.0952106,-0.2909335,-0.000086550},
```

```

158                                     {0.19377028,0.19080467,0.17636244,0.0000523508}};
159 float vec_y0[4] = {0,0,0,0};
160 float ybyf[4] = {0,0,0,0};
161 float Ubyt[4] = {0,0,0,0};
162 float uy = 0.0;
163 float y_k[4] = {0,0,0.015525862,0};
164 float mAy[4][4] = {{0.99999998,0.0024999778,0.000030538312,0},
165                    {-0.000016426937,0.99998222,0.024430649,0},
166                    {-0.0013396075,-0.0014496127,0.99230576,0.0004786864},
167                    {-1.071686,-1.1596902,-6.1553959,-0.61705088}};
168 float mBy[4] = {0,0,0.000044653582,0.035722866};
169 float mCy[4][4] = {{0.99999999,0.0012499889,0.000015269156,0},
170                    {0,0.99999111,0.012215325,0},
171                    {-0.00066980374,-0.0007248063,0.99615288,0.000239343},
172                    {-0.53584299,-0.57984508,-3.077698,0.19147456}};
173 float mDy[4] = {0,0,0.000022326791,0.017861433};
174 float Out_Yy_k_1[4] = {0,0,0,0};
175 float Out_Yy_k[4] = {0,0,0,0};
176 float out_Yy_k_1[4] = {0,0,0,0};
177 float out_Yy_k[4] = {0,0,0,0};
178 }vRy;
179
180 #endif /* PARAMETERS_H */

```

Cada vez que se cambie de una trayectoria lenta a una trayectoria rápida y viceversa, manteniendo el mismo radio  $r_* = 5$  m, es necesario modificar los valores de  $N_p$  (líneas 114 y 149) y los valores de los arreglos  $MExp_xTtr$  y  $MExp_yTtr$  (líneas 118 a 121 y 153 a 156).

#### D.4.2.2 Parámetros para el Pixhawk

En el siguiente código se definen constantes que se utilizan para controlar el brillo del led del Pixhawk y parámetros de registro.

```
1 /*
```

#### D.4. Descripción del código personalizado

---

```
2 *      Author: ANGHEL
3 */
4
5 #ifndef _DEFINES_H
6 #define _DEFINES_H
7
8 # define MAIN_LOOP_MICROS    2500
9
10 #define LED_BRIGHT    0xFF
11 #define LED_MEDIUM    0x80
12 #define LED_DIM        0x11
13 #define LED_OFF        0x00
14
15 #define LOG_TEST_MSG1  0x01
16 #define LOG_TEST_MSG2  0x03
17 #define LOG_TEST_MSG3  0x06
18 #define LOG_TEST_MSG4  0x09
19 #define LOG_TEST_MSG5  0x0C
20 #define LOG_TEST_MSG6  0x0F
21 #define LOG_TEST_MSG7  0x12
22
23 #endif /* DEFINES_H */
```

En la línea 8 se define una constante que se utiliza en la línea 211 de D.4.1.5 para el cálculo del tiempo disponible en el planificador de tareas. Las constantes definidas en las líneas 10 a 11 se utilizan para controlar el brillo del led del Pixhawk. En las líneas 15 a 21 se definen parámetros de registro, esto es, direcciones de memoria para cada uno de los paquetes de datos que se van a guardar en la memoria micro SD (ver por ejemplo las líneas 17 y 142).

Debido a que existen varias tareas que deben realizarse en un corto periodo de tiempo, esto es, la lectura de sensores, comunicación a través del radio, gestión de los actuadores; se utiliza un planificador o programador de tareas. Este se encarga de ejecutar todas las tareas

(funciones) posibles considerando su prioridad y frecuencia dada. El void loop() se encarga de gestionar el tiempo que se le da al programador de tareas.

### D.4.2.3 Lectura del radiocontrol

```
1 /*
2  * radio.pde
3  * Author: ANGHEL BLAS
4  */
5
6 static void Read_radio()
7 {
8     uint16_t radio [7];
9
10    for (uint8_t i=0; i<7; i++){
11        radio [i] = hal.rcin->read(i);
12    }
13
14    rc.rc_Roll = (radio [0]-1498)/3;
15    rc.rc_Pitch = (radio [1]-1498)/3;
16    rc.rc_Yaw = (radio [3]-1499)/3;
17    rc.rc_Throttle = radio [2];
18    rc.Aux.1 = radio [4];
19    rc.Aux.2 = radio [5];
20    rc.Aux.3 = radio [6];
21
22    m1_rc = rc.rc_Throttle - rc.rc_Yaw - rc.rc_Pitch + rc.rc_Roll;
23    m2_rc = rc.rc_Throttle - rc.rc_Yaw + rc.rc_Pitch - rc.rc_Roll;
24    m3_rc = rc.rc_Throttle + rc.rc_Yaw - rc.rc_Pitch - rc.rc_Roll;
25    m4_rc = rc.rc_Throttle + rc.rc_Yaw + rc.rc_Pitch + rc.rc_Roll;
26 }
```

En la línea 6 se especifica el nombre de la tarea tal y como se declara en el planificador de tareas, esto es, **Read\_radio**. En la línea 8 definimos la variable para almacenar las señales

del radiocontrol, de tipo array y formato entero sin signo. En las líneas 10 a 12 se lleva a cabo la lectura de los canales del radiocontrol y el almacenamiento en el array mediante un lazo for. En las líneas 14 a 20 se asignan a nuevas variables los valores de los canales del radio almacenados en el arreglo `radio[i]`, y se hace un mapeo de los valores de los canales roll, pitch y yaw, esto es, `radio[0]`, `radio[1]` y `radio[3]` respectivamente. Esto se hace con el objetivo de leer un valor alrededor de cero cuando las palancas están centradas y al mismo tiempo son escaladas a un valor máximo o mínimo cuando las palancas se mueven a la derecha, izquierda, adelante o atrás. El canal `radio[2]` que corresponde al throttle no necesita escalarse. La suma de las señales que se envían a los motores se definen en las líneas 22 a 25.

##### D.4.2.4 Lectura del GPS

```
254 static void GPS_call()
255 {
256     float dt;
257     static uint32_t last_msg_ms;
258     gps.update();
259     if(last_msg_ms != gps.last_message_time_ms())
260     {
261         last_msg_ms = gps.last_message_time_ms();
262         const Location &loc = gps.location();
263         flag_gps = gps.status();
264     }
265
266     uint32_t currtime = hal.scheduler->millis();
267     dt = (float)(currtime - last_update) / 1000.0f;
268     last_update = currtime;
269     inertial_nav.update(dt);
270
271     if(flag_gps >=3 && entra_home==true)
272     {
273         const Location &loc = gps.location();
```

```
274     ahrs.set_home(loc);
275     compass.set_initial_location(loc.lat, loc.lng);
276     entra_home = false;
277     }
278
279     pos = inertial_nav.get_position()*0.01;
280     vel = inertial_nav.get_velocity()*0.01;
281
282     toshiba_led.update();
283     if(flag_gps >= 3 && pos.x != 0){
284         toshiba_led.set_rgb(LED_MEDIUM, 0, LED_MEDIUM);
285     }
286
287     gps_orig();
288
289 }
```

La línea 254 especifica el nombre de la tarea declarada en el planificador de tareas, la cual permite obtener las señales de posición y velocidad del vehículo, esto es, **GPS\_call**. En las líneas 256 y 257 se definen las variables locales para la delta de tiempo y el mensaje. En el lazo if dado en las líneas 259 a 264 se comprueba que el último mensaje sea diferente al que se está leyendo para volver a actualizar el mensaje, la ubicación y el estado del GPS.

En las líneas 266 a 268 se calcula la delta de tiempo  $dt$ . En 266 se almacena el tiempo actual, en 267 se calcula la delta de tiempo y en 268 el tiempo actual se guarda para ser utilizado en la línea 267 como tiempo en el instante anterior. La delta de tiempo se utiliza en la línea 269 para actualizar los estimados de la velocidad y posición usando la última información de los acelerómetros aumentado con las lecturas del GPS y barómetro.

En las líneas 271 a 277 se define el home, esto es, la posición de vehículo cuando se tienen lecturas correctas del GPS. Esto se ejecuta solo una vez cuando se cumple la condición impuesta en el lazo if. Las líneas 279 y 280 devuelven la posición actual relativa a la ubicación

del home en centímetros y la velocidad<sup>6</sup> actual en cm/s respectivamente. Se multiplican por 0.01 para la conversión a metros. En la línea 282 se actualiza el led toshiba y las líneas 283 a 285 permiten que el led encienda en color magenta cuando se cumple la condición impuesta el lazo if, esto se hace para indicar que se tienen lecturas correctas en posición  $(x, y)$ .

En la línea 287 se hace el llamado de la función **gps\_orig**, esta permite definir arbitrariamente el origen  $(x, y) = (0, 0)$ .

##### D.4.2.5 Lectura del barómetro

```
292 static void Barometro_call()
293 {
294     barometer.update();
295     baro.z_baro = barometer.get_altitude();
296 }
```

En la línea 292 se especifica el nombre de la tarea declarada en el planificador de tareas. En la línea 294 se hace la actualización del sensor para que en la línea 295 se adquiriera la señal de altura del barómetro.

##### D.4.2.6 Guardar datos en la memoria micro SD

```
299 static void SaveMemorySD_call()
300 {
301     Log_Write1();
302     Log_Write2();
303     Log_Write3();
304     Log_Write4();
305     Log_Write5();
306     Log_Write6();
307 }
```

---

<sup>6</sup>X speed cm/s (+ve North), Y speed cm/s (+ve East), Z speed cm/s (+ve up)



En la línea 299 se especifica el nombre de la tarea declarada en el planificador de tareas, esto es, **SaveMemorySD\_call**. En las líneas 301 a 306 se hace el llamado a de las funciones **Log\_Write1** a **Log\_Write1** para escribir los paquetes de datos.

### D.4.2.7 Iniciar ArduPilot

La función **init\_ardupilot** inicia sensores y habilita canales de salida

```
1 /*
2  * Author: ANGHEL
3  */
4
5
6 static void init_ardupilot ()
7 {
8     serial_manager.init_console ();
9     serial_manager.init ();
10    compass.init ();
11    gps.init (NULL, serial_manager);
12    barometer.init ();
13    barometer.calibrate ();
14    toshiba_led.init ();
15    ahrs.set_compass(&compass);
16
17    DataFlash.Init (log_structure ,
18                    sizeof(log_structure)/sizeof(log_structure [0]));
19    if (DataFlash.NeedErase ()) {
20        DataFlash.EraseAll ();
21    }
22    log_num1 = DataFlash.StartNewLog ();
23
24    hal.rcout->set_freq (15,490);
25    hal.rcout->enable_ch (0);
26    hal.rcout->enable_ch (1);
27    hal.rcout->enable_ch (2);
```

```
28     hal.rcout->enable_ch(3);
29 }
```

La línea 8 se inicia la consola, en la línea 9 se inicializan los puertos seriales, en la línea 10 se inicializa el dispositivo brújula. En la línea 11 se inicializa el GPS, en las líneas 12 y 13 se inicializa y calibra el barómetro. En la línea 14 se inicializa el led del Pixhawk. En las líneas 17 a 21 se inicializa la escritura en la memoria micro SD. La línea 22 tiene que ver con el último archivo binario log guardado y el inicio de un nuevo archivo binario log.

En la línea 24 con el número 15 se indica que se utilizarán 4 canales de salida y estos se inicializan a la frecuencia de los controladores electrónicos de velocidad (ESC por sus siglas en inglés) de 490 Hz. En las líneas 25 a 28 se indica con los números del 0 al 3 los canales de salida habilitados.

### D.4.2.8 Fast loop

**fast\_loop** es la función crítica del código principal, en la figura D.14 podemos ver las funciones e instrucciones que deben llamarse a la prioridad más alta. El **fast\_loop** es llamado directamente del void loop() y no del planificador de tareas (scheduler), porque se garantiza la ejecución en cada ciclo.

```
215 // Main loop 400 Hz
216 static void fast_loop()
217 {
218     ahrs.update();
219     Gyro = ins.get_gyro();
220
221     ctrl_FB();
222     obser1();
223     trajectory();
224
225     if(rc.rc_Throttle >= 1150){
226         if(rc.Aux_2 < 1700){
227             m1 = m1.Ctrl + m1.rc;
```

```

228         m2 = m2_Ctrl + m2_rc;
229         m3 = m3_Ctrl + m3_rc;
230         m4 = m4_Ctrl + m4_rc;
231         fg = 2;
232     }
233     else{
234         m1 = m1_Ctrl + m1_rc + m1_obs;
235         m2 = m2_Ctrl + m2_rc + m2_obs;
236         m3 = m3_Ctrl + m3_rc + m3_obs;
237         m4 = m4_Ctrl + m4_rc + m4_obs;
238         fg = 1;
239     }
240 }
241 else{
242     m1 = 1100;
243     m2 = 1100;
244     m3 = 1100;
245     m4 = 1100;
246     fg = 0;
247 }
248 hal.rcout->write(0,m1);
249 hal.rcout->write(1,m2);
250 hal.rcout->write(2,m3);
251 hal.rcout->write(3,m4);
252 }

```

En línea 218 se actualiza `ahrs` para que en la línea 219 se obtengan las velocidades angulares. En las líneas 221 a 223 se llaman las funciones `ctrl_FB`, `obser1` y `trajectory` para el control por realimentación de estado, los filtros Beard-Jones discretos y las trayectorias respectivamente.

En las líneas 225 a 247 se tienen dos lazos if-else anidados. En el lazo interno if-else se calcula la suma de señales sin el filtro Beard-Jones (líneas 227 a 230) y con filtro Beard-Jones (líneas 234 a 237). Estos dos casos dependen de la posición del interruptor auxiliar 2 del

radiocontrol.

`m1_Ctrl` a `m4_Ctrl` es la suma de las señales de control por realimentación de estado y se calcula en la función `ctrl_FB` (ver línea 221). `m1_rc` a `m4_rc` es la suma de las señales del radiocontrol y se calcula en la función `Read_radio` declarada en el planificador de tareas (ver línea 190). `m1_obs` a `m4_obs` es la suma de señales de los filtros Beard-Jones y se calculan en la función `obser1` (ver línea 222).

El lazo if-else externo se introduce por seguridad y depende del throttle del radiocontrol. El rango del throttle es de 1100 a 1900 y corresponden a un valor mínimo y máximo de PWM. Cuando la condición impuesta es menor a 1150, se envía una señal PWM en la que los motores no giran. Cuando el valor del throttle es igual o mayor al valor de 1150, entra el lazo interno if-else descrito anteriormente.

Finalmente, las funciones en las líneas 248 a 251 envían a los pines de salida (0, 1, 2, 3) las suma se señales dadas en (`m1`, `m2`, `m3`, `m4`).

### D.4.3 Funciones de funciones del programa principal

#### D.4.3.1 Generar origen arbitrario

Una vez que el cuadricóptero está listo para volar, lo colocamos en el exterior y energizamos la electrónica. El punto donde se encuentre ubicado el cuadricóptero será el origen en posición ( $x, y, z$ ) cuando se tenga una buena señal de GPS, no importa si mueve en cualquier dirección. Para propósitos prácticos, nos interesa en generar un origen arbitrario en ( $x, y$ ), esto se logra mediante una simple resta de posiciones iniciada por la señal un interruptor del radiocontrol.

```
1 /*
2  * Author: ANGHEL BLAS
3  */
4
5
6 static void gps_orig()
7 {
```

```

8   if(rc.Aux_1 < 1500){
9       if (entra_offset==true){
10          pos_offset = {pos.x, pos.y, 0};
11          pos_offset1 = pos_offset;
12          entra_offset = false;
13      }
14      pos_offset = {vRx.out_Xy_k[0]+pos_offset1.x,
15                  vRy.out_Yy_k[0]+pos_offset1.y, 0};
16      pos_xyz = pos-pos_offset;
17      vel_xyz = {vel.x-vRx.out_Xy_k[1], vel.y-vRy.out_Yy_k[1], vel.z};
18
19      pos_xyz1 = pos-pos_offset1;
20      vel_xyz1 = {vel.x, vel.y, vel.z};
21  }
22  else{
23      entra_offset = true;
24      pos_xyz = {0,0,0};
25      vel_xyz = {0,0,0};
26      pos_offset = {0,0,0};
27      pos_xyz1 = {0,0,0};
28      vel_xyz1 = {0,0,0};
29  }
30 }

```

A grandes razgos este fragmento de código hace lo siguiente: se tiene un if anidado dentro de un lazo if-else, el lazo if-else tiene por condición que el valor del interruptor auxiliar 1 del radiocontrol sea menor a 1500. Si el valor del interruptor auxiliar 1 siempre es mayor a 1500 (se ejecutan las líneas 23 a 28), a la variable booleana de la línea 23 se le asigna true y a las variables definidas como Vector3f de las líneas 24 a 28 se les asigna cero. Lo anterior permite volar de forma manual el cuadricóptero en cualquier dirección.

Cuando el interruptor auxiliar 1 es menor a 1500 (se ejecutan las líneas 9 a 20), el if anidado se ejecuta solo una vez y su función es adquirir el valor de posición  $(x, y)$  (un valor constante). Para esto comprueba que la variable `entra_offset` sea verdadera en la línea 9, se

adquiere la posición en las líneas 10 y 11, finalmente a la variable `entra_offset` se le asigna falso para que en el siguiente ciclo no se cumpla la condición del `if`.

En las líneas 14 y 15 a la variable `pos_offset` se le asigna el valor de la posición adquirida más el valor de la trayectoria calculada ese instante (si el seguimiento de trayectoria está activado se tiene un valor, si el seguimiento está desactivado el valor es cero). En la línea 16 se hace la resta de posición actual medida y la posición adquirida, lo anterior permite generar un nuevo origen arbitrario y el resultado se utiliza en las realimentaciones de estado. En la línea 17 se obtiene el error de la velocidad medida menos la trayectoria de velocidad y se utiliza en las realimentaciones de estado. Las líneas 19 y 20 no consideran las trayectorias y se introducen solamente para propósitos de comparación con las trayectorias programadas.

### D.4.3.2 Paquetes de datos

Para la construcción de los paquetes de datos se realiza lo siguiente: se crean las estructuras de inicialización, las funciones para guardar datos y se definen los encabezados de los paquetes de datos (lo que se verá en Mission Planner). Únicamente se describe la forma de guardar un paquete de datos porque para los demás paquetes la explicación es similar.

```
1 /*
2  *      Author: ANGHEL
3  */
4
5
6 struct PACKED log_Test1 {
7     LOG_PACKET_HEADER;
8     uint32_t tiempo_ms;
9     float Roll, Pitch, Yaw, GyroX, GyroY, GyroZ;
10    float posX, posY, posZ, rateX, rateY, rateZ;
11    };
12
13 //Write a packet
14 static void Log_Write1()
```

```
15 {
16     struct log_Test1 pkt = {
17         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG1),
18         tiempo_ms : hal.scheduler->millis(),
19         Roll      : ahrs.roll,
20         Pitch     : ahrs.pitch,
21         Yaw       : ahrs.yaw,
22         GyroX     : Gyro.x,
23         GyroY     : Gyro.y,
24         GyroZ     : Gyro.z,
25         posX      : pos.x,
26         posY      : pos.y,
27         posZ      : pos.z,
28         rateX     : vel.x,
29         rateY     : vel.y,
30         rateZ     : vel.z
31     };
32     DataFlash.WriteBlock(&pkt, sizeof(pkt));
33 }
34
35
36 struct PACKED log_Test2 {
37     LOG_PACKET_HEADER;
38     float cRoll, cPitch, cYaw, cX, cY, cZ, qxt, qyp, qpsi;
39 };
40
41 //Write a packet
42 static void Log_Write2()
43 {
44     struct log_Test2 pkt = {
45         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG2),
46         cRoll     : Ctrl_Roll,
47         cPitch    : Ctrl_Pitch,
48         cYaw      : Ctrl_Yaw,
```

#### D.4. Descripción del código personalizado

---

```
49         cX         : Ctrl_x ,
50         cY         : Ctrl_y ,
51         cZ         : Ctrl_z ,
52         qxt        : Obsxt1.out_xy_k ,
53         qyp        : Obsyp1.out_yy_k ,
54         qpsi       : Obsp1.out_py_k
55     };
56     DataFlash.WriteBlock(&pkt , sizeof(pkt));
57 }
58
59
60 struct PACKED log_Test3 {
61     LOG_PACKET_HEADER;
62     float tim;
63 };
64
65 //Write a packet
66 static void Log_Write3()
67 {
68     struct log_Test3 pkt = {
69         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG3) ,
70         tim          : 81.45
71     };
72     DataFlash.WriteBlock(&pkt , sizeof(pkt));
73 }
74
75
76 struct PACKED log_Test4 {
77     LOG_PACKET_HEADER;
78     float Posz , ZrefF , Flag;
79 };
80
81 //Write a packet
82 static void Log_Write4()
```



```

83 {
84     struct log_Test4 pkt = {
85         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG4),
86         Posz      : pos_xyz.z,
87         ZrefF     : z_ref_F,
88         Flag      : fg,
89     };
90     DataFlash.WriteBlock(&pkt, sizeof(pkt));
91 }
92
93
94 struct PACKED log_Test5 {
95     LOG_PACKET_HEADER;
96     float T1x, T2Dx, T3th, T4Dth, T1y, T2Dy, T3ph, T4Dph;
97 };
98
99 //Write a packet
100 static void Log_Write5()
101 {
102     struct log_Test5 pkt = {
103         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG5),
104         T1x      : vRx.out_Xy_k[0],
105         T2Dx     : vRx.out_Xy_k[1],
106         T3th     : vRx.out_Xy_k[2],
107         T4Dth    : vRx.out_Xy_k[3],
108         T1y      : vRy.out_Yy_k[0],
109         T2Dy     : vRy.out_Yy_k[1],
110         T3ph     : vRy.out_Yy_k[2],
111         T4Dph    : vRy.out_Yy_k[3]
112     };
113     DataFlash.WriteBlock(&pkt, sizeof(pkt));
114 }
115
116

```

#### D.4. Descripción del código personalizado

---

```
117 struct PACKED log_Test6 {
118     LOG_PACKET_HEADER;
119     float pxT, vxT, theT, DtheT, pyT, vyT, phiT, DphiT;
120 };
121
122 //Write a packet
123 static void Log_Write6()
124 {
125     struct log_Test6 pkt = {
126         LOG_PACKET_HEADER_INIT(LOG_TEST_MSG6),
127         pxT      : pos_xyz1.x,
128         vxT      : vel_xyz1.x,
129         theT     : pitchT,
130         DtheT    : DpitchT,
131         pyT      : pos_xyz1.y,
132         vyT      : vel_xyz1.y,
133         phiT     : rollT,
134         DphiT    : DrollT
135     };
136     DataFlash.WriteBlock(&pkt, sizeof(pkt));
137 }
138
139
140 static const struct LogStructure log_structure [] PROGMEM = {
141     LOG_COMMON_STRUCTURES,
142     { LOG_TEST_MSG1, sizeof(log_Test1),
143     "1", "Iffffffffffff", "t_ms, Roll, Pitch, Yaw, gyrX, gyrY, gyrZ, pX, pY, pZ, vX, vY, vZ" },
144     { LOG_TEST_MSG2, sizeof(log_Test2),
145     "2", "ffffffff", "CRoll, CPitch, CYaw, CX, CY, CZ, q-xt, q-yp, q-p" },
146     { LOG_TEST_MSG3, sizeof(log_Test3),
147     "3", "f", "cont" },
148     { LOG_TEST_MSG4, sizeof(log_Test4),
149     "4", "fff", "Pz, ZdesF, flag" },
150     { LOG_TEST_MSG5, sizeof(log_Test5),
```

```
151 "5", "ffffffff", "T1x,T2Dx,T3th,T4Dth,T1y,T2Dy,T3ph,T4Dph" },
152 { LOG_TEST_MSG6, sizeof(log_Test6),
153 "6", "ffffffff", "PxT,VxT,TheT,DTheT,PyT,VyT,PhiT,DPhiT" },
154 };
155
156 static uint16_t log_num1;
```

Primero se describe la estructura de inicialización. En la línea 6 se define el nombre de la estructura, esto es, `log_Test1`. Con este nombre será llamada en la estructura dentro de la función para guardar datos. El código de la línea 7 se escribe en todas las estructuras de inicialización. En una estructura de datos, los datos pueden tener diferente formato: flotante, entero sin signo, etc. Note que en las líneas 8 a 10, primero se escribe el formato del dato y a continuación el nombre del dato a guardar. Por ejemplo: en la línea 8, en formato entero sin signo de 32 bits se almacena el tiempo, la variable utilizada para esto es `tiempo_ms`.

A continuación se describe la función para guardar datos. En la línea 14 se le asigna un nombre a la función, con este nombre se llama en la línea 301 de la función **SaveMemorySD\_call**. En la línea 16 se hace el llamado de la estructura de inicialización descrito anteriormente. En la línea 17 al paquete de datos se le asigna una dirección de memoria, es decir, `LOG_TEST_MSG1` tiene asignado un valor constante que corresponde a la dirección de memoria. En las líneas 18 a 30 se tienen dos columnas separadas por dos puntos. La primera columna corresponde a los nombres asignados a las variables en la estructura de inicialización. La segunda columna corresponde a las variables globales utilizadas en el programa, estas variables son las que tienen los datos que se desean guardar. La línea 32 corresponde al paquete o bloque de datos a guardar en la memoria micro SD y el tamaño en bytes que ocupa `pkt`.

Y finalmente se define los encabezados de los paquetes de datos. Lo anterior se lleva a cabo en las líneas 140 a 154. En la línea 142 se declara la dirección de memoria para el paquete de datos y se obtiene el tamaño en bytes que ocupa la variable `log_Test1`. En la línea 143 el número uno corresponde al nombre del paquete de datos, le sigue el formato

de los datos y finalmente el nombre de los datos almacenados (estos nombres son los que se mostrarán en el Mission Planner).

##### D.4.3.3 Realimentaciones de estado

En el siguiente código se describe el control por realimentación de estado para el cuadricóptero.

```
1 /*
2 *      Author: ANGHEL
3 */
4
5
6 static void ctrl_FB()
7 {
8     roll = ahrs.roll - vRy.out_Yy_k[2];
9     Droll = Gyro.x - vRy.out_Yy_k[3];
10    pitch = ahrs.pitch - vRx.out_Xy_k[2];
11    Dpitch = Gyro.y - vRx.out_Xy_k[3];
12    yaw = ahrs.yaw;
13    Dyaw = Gyro.z;
14
15    Ctrl_Roll = sat_signal(- PDroll.Kp*roll - PDroll.Kd*Droll,170,-170);
16    Ctrl_Pitch = sat_signal(- PDpitch.Kp*pitch - PDpitch.Kd*Dpitch,170,-170);
17    Ctrl_Yaw = sat_signal(- PDyaw.Kp*yaw - PDyaw.Kd*Dyaw,170,-170);
18
19    if (rc.Aux_1 < 1500){
20        if (Adq_z_ref == true){
21            z_ref_F = pos.z;
22            Adq_z_ref = false;
23        }
24        px = pos_xyz.x;
25        Dx = vel_xyz.x;
26        py = pos_xyz.y;
27        Dy = vel_xyz.y;
28
```

```

29     pitchT = ahrs.pitch;
30     DpitchT = Gyro.y;
31     rollT = ahrs.roll;
32     DrollT = Gyro.x;
33
34     Ctrl_z = - PDz1.Kp*(pos.z - z_ref_F) - PDz1.Kd*vel.z;
35     Ctrl_z = sat_signal(Ctrl_z,95,-95);
36
37     Ctrl_x = sat_signal(- PDx.Kp*px - PDx.Kd*Dx,80,-80);
38     Ctrl_y = sat_signal(- PDy.Kp*py - PDy.Kd*Dy,80,-80);
39 }
40 else{
41     Adq_z_ref = true;
42     Ctrl_x = 0;
43     Ctrl_y = 0;
44     Ctrl_z = 0;
45     z_ref_F = 0;
46     err_x = 0;
47     pitchT = 0;
48     DpitchT = 0;
49     rollT = 0;
50     DrollT = 0;
51 }
52
53 m1_Ctrl = - Ctrl_Roll + Ctrl_Pitch + Ctrl_Yaw + Ctrl_z + Ctrl_x - Ctrl_y;
54 m2_Ctrl =  Ctrl_Roll - Ctrl_Pitch + Ctrl_Yaw + Ctrl_z - Ctrl_x + Ctrl_y;
55 m3_Ctrl =  Ctrl_Roll + Ctrl_Pitch - Ctrl_Yaw + Ctrl_z + Ctrl_x + Ctrl_y;
56 m4_Ctrl = - Ctrl_Roll - Ctrl_Pitch - Ctrl_Yaw + Ctrl_z - Ctrl_x - Ctrl_y;
57 }

```

En las líneas 8 a 13 se tienen las posiciones y velocidades angulares cuando el seguimiento de trayectoria está desactivado. Cuando el seguimiento esta activado, se obtiene el error en seguimiento. En ambos casos el resultado obtenido se utiliza en las realimentaciones de estado. En las líneas 15 a 17 se calculan las realimentaciones de estado en orientación. Se

ha comprobado de manera experimental que las señales de realimentación no sobrepasan los límites de la función de saturación (ver líneas de código 310 a 315 de la función **sat\_signal**).

En las líneas 19 a 51 se tiene un lazo if anidado dentro de un lazo if-else. El lazo if-else tiene por condición que el valor del interruptor auxiliar 1 del radiocontrol sea menor a 1500. Si el valor del interruptor auxiliar 1 siempre es mayor a 1500 (se ejecutan las líneas 41 a 50), a la variable booleana de la línea 41 se le asigna true y a las variables definidas como Vector3f o float de las líneas 42 a 50 se les asigna cero.

Cuando el interruptor auxiliar 1 es menor a 1500 (se ejecutan las líneas 20 a 38), el if anidado se ejecuta solo una vez y su función es adquirir el valor de altura  $z$  (un valor constante). Para esto comprueba que la variable `Adq_z_ref` sea verdadera en la línea 20, se adquiere el valor de la altura en las líneas 21 y finalmente a la variable `Adq_z_ref` se le asigna falso para que en el siguiente ciclo no se cumpla la condición del if.

En las líneas 24 a 27 se tienen posiciones y velocidades cuando el seguimiento de trayectoria está desactivado y los errores de posición y velocidad cuando el seguimiento está activado (ver líneas de código 14 a 16 de la función **gps\_origen**).

Las líneas 29 a 32 se introducen solamente para propósitos de comparación con las trayectorias programadas.

En las líneas 34 y 35 se calcula la señal de control por realimentación de estado en altura y se utiliza la función de saturación. En las líneas 37 y 38 se calculan las señales de control por realimentación de estado para  $x$  y  $y$ . En ambos casos se utiliza la función de saturación.

Finalmente, en las líneas 53 a 56 se obtiene la suma de las señales de control por realimentación que se van a enviar a cada motor.

#### D.4.3.4 Filtros Beard-Jones

El filtro Beard-Jones sin deriva es

$$\begin{aligned}\frac{d}{dt}\bar{w}_{df,x} &= \left(\bar{A}_{\bar{K}_x} + (1 - \epsilon)\bar{B}_x G_x^\ell \bar{C}_x\right)\bar{w}_{df,x} - \left(\bar{K}_x + (1 - \epsilon)\bar{B}_x G_x^\ell\right)x, \\ \bar{u}_{df,x} &= (1 - \epsilon)G_x^\ell (\bar{C}_x \bar{w}_{df,x} - x),\end{aligned}\tag{D.1}$$

Aproximando la derivada con  $w_{\text{diff}} = \frac{1}{T_s}[w(k+1) - w(k)]$ , el estimador se reescribe en la forma discreta como

$$\begin{aligned}\bar{w}_x(k+1) &= A_x \bar{w}_x(k) + B_x x(k), \\ \bar{u}_x(k) &= C_x \bar{w}_x(k) + D_x x(k),\end{aligned}\tag{D.2}$$

A continuación se describe el código en el que se implementa el estimador dado en (D.2) para los subsistemas  $x - \theta$ ,  $y - \phi$  y  $\psi$ .

```

1 /*
2 *      Author: ANGHEL
3 */
4
5
6 static void obser1()
7 {
8     if(rc.Aux_1 < 1500){
9         Obsxt1.u_x = pos_xyz.x;
10        Obsyp1.u_y = pos_xyz.y;
11        Obsp1.u_p = ahrs.yaw;
12
13        Obsxt1.Out_xy_k = 0;
14        Obsxt1.out_xy_k = 0;
15
16        Obsyp1.Out_yy_k = 0;
17        Obsyp1.out_yy_k = 0;
18
19        Obsp1.Out_py_k = 0;
20        Obsp1.out_py_k = 0;
21
22        for(int i = 0; i < 4; i++){
23            Obsxt1.Out_xx_k_1[i] = 0;
24            Obsyp1.Out_yy_k_1[i] = 0;
25        }
26        Obsxt1.Out_xx_k_1[0] = Obsxt1.Aex[0][0]*Obsxt1.vbcix[0] +
27                               Obsxt1.Aex[0][1]*Obsxt1.vbcix[1] +

```

```

28             Obsxt1.Aex[0][2]*Obsxt1.vbcix[2] +
29             Obsxt1.Aex[0][3]*Obsxt1.vbcix[3] +
30             Obsxt1.Bex[0]*Obsxt1.u_x;
31     Obsxt1.Out_xx_k_1[1] = Obsxt1.Aex[1][0]*Obsxt1.vbcix[0] +
32             Obsxt1.Aex[1][1]*Obsxt1.vbcix[1] +
33             Obsxt1.Aex[1][2]*Obsxt1.vbcix[2] +
34             Obsxt1.Aex[1][3]*Obsxt1.vbcix[3] +
35             Obsxt1.Bex[1]*Obsxt1.u_x;
36     Obsxt1.Out_xx_k_1[2] = Obsxt1.Aex[2][0]*Obsxt1.vbcix[0] +
37             Obsxt1.Aex[2][1]*Obsxt1.vbcix[1] +
38             Obsxt1.Aex[2][2]*Obsxt1.vbcix[2] +
39             Obsxt1.Aex[2][3]*Obsxt1.vbcix[3] +
40             Obsxt1.Bex[2]*Obsxt1.u_x;
41     Obsxt1.Out_xx_k_1[3] = Obsxt1.Aex[3][0]*Obsxt1.vbcix[0] +
42             Obsxt1.Aex[3][1]*Obsxt1.vbcix[1] +
43             Obsxt1.Aex[3][2]*Obsxt1.vbcix[2] +
44             Obsxt1.Aex[3][3]*Obsxt1.vbcix[3] +
45             Obsxt1.Bex[3]*Obsxt1.u_x;
46
47     Obsyp1.Out_yy_k_1[0] = Obsyp1.Aey[0][0]*Obsyp1.vbciy[0] +
48             Obsyp1.Aey[0][1]*Obsyp1.vbciy[1] +
49             Obsyp1.Aey[0][2]*Obsyp1.vbciy[2] +
50             Obsyp1.Aey[0][3]*Obsyp1.vbciy[3] +
51             Obsyp1.Bey[0]*Obsyp1.u_y;
52     Obsyp1.Out_yy_k_1[1] = Obsyp1.Aey[1][0]*Obsyp1.vbciy[0] +
53             Obsyp1.Aey[1][1]*Obsyp1.vbciy[1] +
54             Obsyp1.Aey[1][2]*Obsyp1.vbciy[2] +
55             Obsyp1.Aey[1][3]*Obsyp1.vbciy[3] +
56             Obsyp1.Bey[1]*Obsyp1.u_y;
57     Obsyp1.Out_yy_k_1[2] = Obsyp1.Aey[2][0]*Obsyp1.vbciy[0] +
58             Obsyp1.Aey[2][1]*Obsyp1.vbciy[1] +
59             Obsyp1.Aey[2][2]*Obsyp1.vbciy[2] +
60             Obsyp1.Aey[2][3]*Obsyp1.vbciy[3] +
61             Obsyp1.Bey[2]*Obsyp1.u_y;

```



```

62     Obsyp1.Out_yy_k_1[3] = Obsyp1.Aey[3][0]*Obsyp1.vbc iy [0] +
63                             Obsyp1.Aey[3][1]*Obsyp1.vbc iy [1] +
64                             Obsyp1.Aey[3][2]*Obsyp1.vbc iy [2] +
65                             Obsyp1.Aey[3][3]*Obsyp1.vbc iy [3] +
66                             Obsyp1.Bey[3]*Obsyp1.u_y;
67
68     Obsxt1.Out_xy_k = Obsxt1.Crx[0]*Obsxt1.vbc ix [0] +
69                             Obsxt1.Crx[1]*Obsxt1.vbc ix [1] +
70                             Obsxt1.Crx[2]*Obsxt1.vbc ix [2] +
71                             Obsxt1.Crx[3]*Obsxt1.vbc ix [3] +
72                             Obsxt1.Drx*Obsxt1.u_x;
73     Obsxt1.out_xy_k = Obsxt1.rhoX*Obsxt1.GfKlx*Obsxt1.Out_xy_k;
74
75     Obsyp1.Out_yy_k = Obsyp1.Cry[0]*Obsyp1.vbc iy [0] +
76                             Obsyp1.Cry[1]*Obsyp1.vbc iy [1] +
77                             Obsyp1.Cry[2]*Obsyp1.vbc iy [2] +
78                             Obsyp1.Cry[3]*Obsyp1.vbc iy [3] +
79                             Obsyp1.Dry*Obsyp1.u_y;
80     Obsyp1.out_yy_k = Obsyp1.rhoY*Obsyp1.GfKly*Obsyp1.Out_yy_k;
81
82     for(int i = 0; i < 2; i++){
83         Obsp1.Out_pp_k_1[i] = 0;
84         Obsp1.out_pp_k_1[i] = 0;
85         for(int j = 0; j < 2; j++){
86             Obsp1.Out_pp_k_1[i] += Obsp1.sysdRpA[i][j]*Obsp1.vbc ip [j];
87         }
88         Obsp1.out_pp_k_1[i] = Obsp1.Out_pp_k_1[i] +
89                                 (Obsp1.sysdRpB[i]*Obsp1.u_p);
90         Obsp1.Out_py_k += Obsp1.sysdRpC[i]*Obsp1.vbc ip [i];
91     }
92     Obsp1.out_py_k = Obsp1.rhoP*Obsp1.GfKlp*(Obsp1.Out_py_k +
93                                         Obsp1.sysdRpD*Obsp1.u_p);
94
95     Obsxt1.vbc ix [0] = Obsxt1.Out_xx_k_1[0];

```

```

96     Obsxt1.vbcix [1] = Obsxt1.Out_xx_k-1 [1];
97     Obsxt1.vbcix [2] = Obsxt1.Out_xx_k-1 [2];
98     Obsxt1.vbcix [3] = Obsxt1.Out_xx_k-1 [3];
99
100    Obsyp1.vbciy [0] = Obsyp1.Out_yy_k-1 [0];
101    Obsyp1.vbciy [1] = Obsyp1.Out_yy_k-1 [1];
102    Obsyp1.vbciy [2] = Obsyp1.Out_yy_k-1 [2];
103    Obsyp1.vbciy [3] = Obsyp1.Out_yy_k-1 [3];
104
105    for (int j=0; j<2; j++){
106        Obsp1.vbcip [j] = Obsp1.out_pp_k-1 [j];
107    }
108
109    m1_obs =  Obsxt1.out_xy_k - Obsyp1.out_yy_k + Obsp1.out_py_k;
110    m2_obs = - Obsxt1.out_xy_k + Obsyp1.out_yy_k + Obsp1.out_py_k;
111    m3_obs =  Obsxt1.out_xy_k + Obsyp1.out_yy_k - Obsp1.out_py_k;
112    m4_obs = - Obsxt1.out_xy_k - Obsyp1.out_yy_k - Obsp1.out_py_k;
113    }
114    else{
115        m1_obs = 0;
116        m2_obs = 0;
117        m3_obs = 0;
118        m4_obs = 0;
119        Obsxt1.vbcix [0] = 1.1435447;
120        Obsxt1.vbcix [1] = -1.253961;
121        Obsxt1.vbcix [2] = 0.11823256;
122        Obsxt1.vbcix [3] = -0.0078162569;
123        Obsxt1.out_xy_k = 0;
124
125        Obsyp1.vbciy [0] = -1.007481;
126        Obsyp1.vbciy [1] = 1.1047346;
127        Obsyp1.vbciy [2] = -0.10413208;
128        Obsyp1.vbciy [3] = 0.0068785223;
129        Obsyp1.out_yy_k = 0;

```

```

130
131     Obsp1.vbcip [0] = -1.3269236;
132     Obsp1.vbcip [1] = 1.4141901;
133     Obsp1.out_py_k = 0;
134     }
135 }
```

El código utiliza un lazo if-else que tiene por condición el valor del interruptor auxiliar 1 del radiocontrol. Si el valor del interruptor auxiliar 1 siempre es mayor a 1500 (se ejecutan las líneas 115 a 133) los estimadores están desactivados.

Cuando el interruptor auxiliar 1 es menor a 1500 (se ejecutan las líneas 9 a 112) se activa los estimadores. Las líneas 9 a 11 corresponden a la entrada de los estimadores sin deriva, esto es,  $x(k)$ ,  $y(k)$  y  $\psi(k)$ . En las líneas 14, 17 y 20 se ponen a cero las salidas de los estimadores sin deriva, esto es,  $\bar{u}_x(k) = 0$ ,  $\bar{u}_y(k) = 0$  y  $\bar{u}_\psi(k) = 0$ . En las líneas 13, 16 y 19 se ponen a cero salidas auxiliares que permiten el cálculo de las salidas de los estimadores (líneas 14, 17 y 20).

El ciclo for de las líneas 22 a 25 pone a cero cada una de cuatro salidas que forman los vectores  $\bar{w}_x(k+1)$ ,  $\bar{w}_y(k+1)$  y  $\bar{w}_\psi(k+1)$ . En las líneas 26 a 66 se calcula cada uno de los términos de manera explícita para los subsistemas  $x - \theta$  y  $y - \phi$ , esto es,  $A_x \bar{w}_x(k) + B_x x(k)$  y  $A_y \bar{w}_y(k) + B_y y(k)$ . En las líneas 68 a 80 se calculan las salidas de los estimadores sin deriva ( $\bar{u}_x(k) = C_x \bar{w}_x(k) + D_x x(k)$  y  $\bar{u}_y(k) = C_y \bar{w}_y(k) + D_y y(k)$ ) con la ayuda de las salidas auxiliares.

A diferencia del cálculo de las salidas de los subsistemas  $x - \theta$  y  $y - \phi$  que se hizo de forma explícita, en las líneas 82 a 93 se calcula la salida  $\bar{u}_\psi(k)$  usando ciclos for anidados.

En las líneas 95 a 107 el valor de la salida actual calculada en las líneas 26 a 66 y línea 88 se asigna a la variable del lado izquierdo, para que en el siguiente ciclo se utilice como el valor anterior en las líneas 26 a 79 y líneas 86 y 90.

Finalmente en las líneas 109 a 112 se calcula la suma de señales de los filtros sin deriva que se enviarán a cada uno de los motores.

### D.4.3.5 Generación de trayectorias

El código para las generación de trayectorias se describirá de manera general.

```
138 static void trajectory ()
139 {
140
141     uint32_t tpo_final1 = vRx.t_hover + (uint32_t)(vRx.Np/0.0025);
142     uint32_t tpo_final2 = vRx.t_hover + (uint32_t)(vRy.Np/0.0025);
143
144     if (rc.Aux_1 < 1500){
145         count0 += 1;
146         if (count0 < vRx.t_hover){
147             vRx.out_Xy_k[0] = 0;
148             vRx.out_Xy_k[1] = 0;
149             vRx.out_Xy_k[2] = 0;
150             vRx.out_Xy_k[3] = 0;
151
152             vRy.out_Yy_k[0] = 0;
153             vRy.out_Yy_k[1] = 0;
154             vRy.out_Yy_k[2] = 0;
155             vRy.out_Yy_k[3] = 0;
156         }
157     else if(count0 >= vRx.t_hover && count0 < tpo_final1){
158         tt7 = 1.5;
159         count1 += 1;
160         t_t1 = count1*0.0025;
161         count2 += 1;
162         t_t2 = count2*0.0025;
163
164         if(t_t1 >= 1){
165             count1 = 0;
166             t_t1 = 0.0;
167             if(ct3 < vRx.Np){
168                 ct3 += 1;
```

```

169         }
170     }
171
172 float Ts1 = (2*M_PI_F)/(vRx.Np*vRx.Ttr);
173 float Ts2 = (2*M_PI_F)/(vRy.Np*vRy.Ttr);
174
175 if (t_t2 < vRx.Np*vRx.Ttr){
176     vRx.vect_x1 [0] = (vRx.xxf/2)*(sin(Ts1*(ct3 + 1)));
177     vRx.vect_x1 [1] = (vRx.xxf/2)*Ts1*(cos(Ts1*(ct3 + 1)));
178     vRx.vect_x1 [2] = (vRx.xxf/2)*(1/param.g)*(pow(Ts1,2))*(sin(Ts1*(ct3+1)));
179     vRx.vect_x1 [3] = (vRx.xxf/2)*(1/param.g)*(pow(Ts1,3))*(cos(Ts1*(ct3+1)));
180
181     vRy.vect_y1 [0] = (vRy.yyf/2)*(1-cos(Ts2*(ct3 + 1)));
182     vRy.vect_y1 [1] = (vRy.yyf/2)*Ts2*(sin(Ts2*(ct3 + 1)));
183     vRy.vect_y1 [2] = (vRy.yyf/2)*(1/param.g)*(pow(Ts2,2))*(cos(Ts2*(ct3+1)));
184     vRy.vect_y1 [3] = (vRy.yyf/2)*(-1/param.g)*(pow(Ts2,3))*(sin(Ts2*(ct3+1)));
185 }
186 else{
187     vRx.vect_x1 [0] = (vRx.xxf/2)*(sin(Ts1*(ct3)));
188     vRx.vect_x1 [1] = (vRx.xxf/2)*Ts1*(cos(Ts1*(ct3)));
189     vRx.vect_x1 [2] = (vRx.xxf/2)*(1/param.g)*(pow(Ts1,2))*(sin(Ts1*(ct3)));
190     vRx.vect_x1 [3] = (vRx.xxf/2)*(1/param.g)*(pow(Ts1,3))*(cos(Ts1*(ct3)));
191
192     vRy.vect_y1 [0] = (vRy.yyf/2)*(1-cos(Ts2*(ct3)));
193     vRy.vect_y1 [1] = (vRy.yyf/2)*Ts2*(sin(Ts2*(ct3)));
194     vRy.vect_y1 [2] = (vRy.yyf/2)*(1/param.g)*(pow(Ts2,2))*(cos(Ts2*(ct3)));
195     vRy.vect_y1 [3] = (vRy.yyf/2)*(-1/param.g)*(pow(Ts2,3))*(sin(Ts2*(ct3)));
196 }
197     vRx.vect_x0 [0] = (vRx.xxf/2)*(sin(Ts1*(ct3)));
198     vRx.vect_x0 [1] = (vRx.xxf/2)*Ts1*(cos(Ts1*(ct3)));
199     vRx.vect_x0 [2] = (vRx.xxf/2)*(1/param.g)*(pow(Ts1,2))*(sin(Ts1*(ct3)));
200     vRx.vect_x0 [3] = (vRx.xxf/2)*(1/param.g)*(pow(Ts1,3))*(cos(Ts1*(ct3)));
201
202     vRy.vect_y0 [0] = (vRy.yyf/2)*(1-cos(Ts2*(ct3)));

```

#### D.4. Descripción del código personalizado

---

```
203     vRy.vect_y0 [1] = (vRy.yyf/2)*Ts2*(sin(Ts2*(ct3)));
204     vRy.vect_y0 [2] = (vRy.yyf/2)*(1/param.g)*(pow(Ts2,2))*(cos(Ts2*(ct3)));
205     vRy.vect_y0 [3] = (vRy.yyf/2)*(-1/param.g)*(pow(Ts2,3))*(sin(Ts2*(ct3)));
206
207
208     vRx.Ubxt [0] = 556.92226*exp(3214.2856*t_t1 - 3214.2856) -
209                    2266.3722*exp(1.9109646*t_t1 - 1.9109646) +
210                    1957.0556*cos(1.5852486*t_t1 -
211                    1.5852486)*exp(0.95359452*t_t1 - 0.95359452) -
212                    182.19143*exp(0.95359452*t_t1 -
213                    0.95359452)*sin(1.5852486*t_t1 - 1.5852486);
214     vRx.Ubxt [1] = 1154.5899*exp(1.9109646*t_t1 - 1.9109646) -
215                    333.57756*exp(3214.2856*t_t1 - 3214.2856) -
216                    1019.7503*cos(1.5852486*t_t1 -
217                    1.5852486)*exp(0.95359452*t_t1 - 0.95359452) +
218                    171.60426*exp(0.95359452*t_t1 -
219                    0.95359452)*sin(1.5852486*t_t1 - 1.5852486);
220     vRx.Ubxt [2] = 2233.9667*exp(1.9109646*t_t1 - 1.9109646) -
221                    817.34859*exp(3214.2856*t_t1 - 3214.2856) -
222                    1652.3768*cos(1.5852486*t_t1 -
223                    1.5852486)*exp(0.95359452*t_t1 - 0.95359452) +
224                    531.32007*exp(0.95359452*t_t1 -
225                    0.95359452)*sin(1.5852486*t_t1 - 1.5852486);
226     vRx.Ubxt [3] = 90.360938*exp(3214.2856*t_t1 - 3214.2856) -
227                    0.69391626*exp(1.9109646*t_t1 - 1.9109646) +
228                    0.51310509*cos(1.5852486*t_t1 -
229                    1.5852486)*exp(0.95359452*t_t1 - 0.95359452) -
230                    0.16513669*exp(0.95359452*t_t1 -
231                    0.95359452)*sin(1.5852486*t_t1 - 1.5852486);
232
233
234     vRy.Ubyt [0] = 2266.0053*exp(1.9109645*t_t1 - 1.9109645) -
235                    556.80465*exp(3358.2089*t_t1 - 3358.2089) -
236                    1956.7843*cos(1.5852486*t_t1 -
```

```

237         1.5852486)*exp(0.95359453*t_t1 - 0.95359453) +
238         182.10414*exp(0.95359453*t_t1 -
239         0.95359453)*sin(1.5852486*t_t1 - 1.5852486);
240     vRy.Ubyt[1] = 333.51151*exp(3358.2089*t_t1 - 3358.2089) -
241         1154.4003*exp(1.9109645*t_t1 - 1.9109645) +
242         1019.6138*cos(1.5852486*t_t1 -
243         1.5852486)*exp(0.95359453*t_t1 - 0.95359453) -
244         171.55439*exp(0.95359453*t_t1 -
245         0.95359453)*sin(1.5852486*t_t1 - 1.5852486);
246     vRy.Ubyt[2] = 2233.5791*exp(1.9109645*t_t1 - 1.9109645) -
247         817.19758*exp(3358.2089*t_t1 - 3358.2089) -
248         1652.1118*cos(1.5852486*t_t1 -
249         1.5852486)*exp(0.95359453*t_t1 - 0.95359453) +
250         531.21432*exp(0.95359453*t_t1 -
251         0.95359453)*sin(1.5852486*t_t1 - 1.5852486);
252     vRy.Ubyt[3] = 90.345434*exp(3358.2089*t_t1 - 3358.2089) -
253         0.66410663*exp(1.9109645*t_t1 - 1.9109645) +
254         0.49107577*cos(1.5852486*t_t1 -
255         1.5852486)*exp(0.95359453*t_t1 - 0.95359453) -
256         0.15803464*exp(0.95359453*t_t1 -
257         0.95359453)*sin(1.5852486*t_t1 - 1.5852486);
258
259     vRx.ux = 0.0;
260     vRy.uy = 0.0;
261
262     for(int i = 0; i < 4; i++){
263         vRx.vec_x0[i] = 0.0;
264         vRy.vec_y0[i] = 0.0;
265         for(int j = 0; j < 4; j++){
266             vRx.vec_x0[i] += vRx.MExpTtr[i][j]*vRx.vect_x0[j];
267
268             vRy.vec_y0[i] += vRy.MExpyTtr[i][j]*vRy.vect_y0[j];
269         }
270     vRx.xbxf[i] = vRx.vect_x1[i] - vRx.vec_x0[i];

```

```

271         vRx.ux += vRx.Ubxt[i]*vRx.xbxf[i];
272
273         vRy.ybyf[i] = vRy.vect_y1[i] - vRy.vec_y0[i];
274         vRy.uy += vRy.Ubyt[i]*vRy.ybyf[i];
275     }
276
277     for(int i = 0; i < 4; i++){
278         vRx.Out_Xx_k_1[i] = 0.0;
279         vRx.out_Xx_k_1[i] = 0.0;
280         vRx.Out_Xy_k[i] = 0.0;
281         vRx.out_Xy_k[i] = 0.0;
282
283         vRy.Out_Yy_k_1[i] = 0.0;
284         vRy.out_Yy_k_1[i] = 0.0;
285         vRy.Out_Yy_k[i] = 0.0;
286         vRy.out_Yy_k[i] = 0.0;
287         for(int j = 0; j < 4; j++){
288             vRx.Out_Xx_k_1[i] += vRx.mAx[i][j]*vRx.x_k[j];
289             vRx.Out_Xy_k[i] += vRx.mCx[i][j]*vRx.x_k[j];
290
291             vRy.Out_Yy_k_1[i] += vRy.mAy[i][j]*vRy.y_k[j];
292             vRy.Out_Yy_k[i] += vRy.mCy[i][j]*vRy.y_k[j];
293
294         }
295         vRx.out_Xx_k_1[i] = vRx.Out_Xx_k_1[i] + vRx.mBx[i]*vRx.ux;
296         vRx.out_Xy_k[i] = vRx.Out_Xy_k[i] + vRx.mDx[i]*vRx.ux;
297
298         vRy.out_Yy_k_1[i] = vRy.Out_Yy_k_1[i] + vRy.mBy[i]*vRy.uy;
299         vRy.out_Yy_k[i] = vRy.Out_Yy_k[i] + vRy.mDy[i]*vRy.uy;
300     }
301
302     for(int j=0; j<4; j++){
303         vRx.x_k[j] = vRx.out_Xx_k_1[j];
304

```



```

305             vRy.y_k[j] = vRy.out_Yy_k-1[j];
306         }
307     }
308     else{
309         vRx.out_Xy_k[0] = 0;
310         vRx.out_Xy_k[1] = 0;
311         vRx.out_Xy_k[2] = 0;
312         vRx.out_Xy_k[3] = 0;
313
314         vRy.out_Yy_k[0] = 0;
315         vRy.out_Yy_k[1] = 0;
316         vRy.out_Yy_k[2] = 0;
317         vRy.out_Yy_k[3] = 0;
318     }
319 }
320 else{
321     vRx.out_Xy_k[0] = 0;
322     vRx.out_Xy_k[1] = 0;
323     vRx.out_Xy_k[2] = 0;
324     vRx.out_Xy_k[3] = 0;
325
326     vRy.out_Yy_k[0] = 0;
327     vRy.out_Yy_k[1] = 0;
328     vRy.out_Yy_k[2] = 0;
329     vRy.out_Yy_k[3] = 0;
330
331     count0 = 0;
332     count1 = 0;
333     count2 = 0;
334     ct3 = 0;
335     t_t1 = 0.0;
336     t_t2 = 0.0;
337     vRx.x_k[0] = 0;
338     vRx.x_k[1] = 0.87266463;

```

#### D.4. Descripción del código personalizado

---

```
339     vRx.x_k [2] = 0;
340     vRx.x_k [3] = 0.0027097742;
341
342     vRy.y_k [0] = 0;
343     vRy.y_k [1] = 0;
344     vRy.y_k [2] = 0.015525862;
345     vRy.y_k [3] = 0;
346     }
347 }
```

Para la generación de las trayectorias se tiene un lazo if-else. Este lazo tiene por condición que el valor del interruptor auxiliar 1 del radiocontrol sea menor a 1500. Si el valor del interruptor auxiliar 1 siempre es mayor a 1500 (se ejecutan las líneas 321 a 345). Es aquí donde se ponen a cero los contadores utilizados en la generación de la escala de tiempo para las trayectorias. También se ponen a cero cada una de las variables asignadas a las trayectorias. Y se pone el estado inicial para poder repetir las trayectorias. Cuando el interruptor auxiliar 1 es menor a 1500 (se ejecutan las líneas 145 a 319). Dentro de esta fracción de código se tienen lazos if-elseif-else, if-else, if anidados y for anidados.

Primero se calcula el tiempo final, este es la suma del tiempo de vuelo estacionario más el tiempo que toma el seguimiento las trayectorias. Lo anterior se realiza en las líneas 141 y 142. El contador count0 de la línea 145 inicia cuando se cumple la condición del valor del interruptor auxiliar 1 menor a 1500. Determinados valores de este contador se utilizan en el lazo if-elseif-else (líneas 144 a 319). lo anterior activa cada una de las etapas que intervienen en el seguimiento de la trayectoria.

En las líneas 147 a 155 de lazo if-elseif-else, el cuadricóptero permanece en vuelo estacionario y las variables para las trayectorias se ponen a cero, es decir, las trayectorias están desactivadas. El tiempo de vuelo estacionario está determinado por la variable t\_hover.

En las líneas 158 a 307 de lazo if-elseif-else, se generan las trayectorias. En las líneas 158 a 170 se crean contadores y senales utilizadas en la generación de las trayectorias. En las líneas 172 a 205 se generan los perfiles de las trayectorias, de estas tomaremos los puntos

estacionarios locales. En las líneas 259 a 275 se generan las entradas de control óptimas. En las líneas 277 a 306 se inyectan la entradas de control óptimas a los sistemas lineales discretos y se obtienen las trayectorias para el seguimiento de trayectoria.

En las líneas 309 a 317 de lazo if-elseif-else, se ejecutan cuando el tiempo para la generación de las trayectorias ha finalizado. De nuevo las variables para las trayectorias se ponen a cero, es decir, las trayectorias están desactivadas.

# Apéndice E

## Publicaciones

## ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect

Journal of the Franklin Institute xxx (xxxx) xxx

[www.elsevier.com/locate/jfranklin](http://www.elsevier.com/locate/jfranklin)

## Robust structural feedback linearization based on the nonlinearities rejection

M. Bonilla<sup>a,\*</sup>, L.A. Blas<sup>a,b</sup>, V. Azhmyakov<sup>c</sup>, M. Malabre<sup>d</sup>, S. Salazar<sup>e</sup>

<sup>a</sup>Control Automático, CINVESTAV-CNRS IPN, CDMX, México

<sup>b</sup>CONACyT, México

<sup>c</sup>Department of Mathematical Sciences, Universidad EAFIT, Medellin, Colombia

<sup>d</sup>CNRS, LS2N, UMR 6004, Nantes, France

<sup>e</sup>Sistemas Autónomos de Navegación Aérea y Submarina, CINVESTAV-CNRS IPN, CDMX, México

Received 1 February 2019; received in revised form 22 August 2019; accepted 13 November 2019

Available online xxx

### Abstract

In this paper, we consider a class of affine control systems and propose a new structural feedback linearization technique. This relatively simple approach involves a generic linear-type control scheme and follows the classic failure detection methodology. The robust linearization idea proposed in this contribution makes it possible an effective rejection of nonlinearities that belong to a specific class of functions. The nonlinearities under consideration are interpreted here as specific signals that affect the initially given systems dynamics. The implementability and efficiency of the proposed robust control methodology is illustrated via the attitude control of a Planar Vertical Take Off Landing (PVTOL) system.

© 2019 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.

### Notation

The following notation is used through this paper.

- Script capitals  $\mathcal{V}$ ,  $\mathcal{W}$ , ... denote finite dimensional linear spaces with elements  $v$ ,  $w$ , ...  
The expression  $\mathcal{V} \approx \mathcal{W}$  stands for  $\dim(\mathcal{V}) = \dim(\mathcal{W})$ . Moreover, when  $\mathcal{V} \subset \mathcal{W}$ ,  $\frac{\mathcal{W}}{\mathcal{V}}$

\* Corresponding author.

E-mail address: [mbonilla@cinvestav.mx](mailto:mbonilla@cinvestav.mx) (M. Bonilla).

<https://doi.org/10.1016/j.jfranklin.2019.11.044>

0016-0032/© 2019 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

or  $\mathcal{W}/\mathcal{V}$  stands for the quotient space  $\mathcal{W}$  modulo  $\mathcal{V}$ . Next,  $\mathcal{V}^\kappa$  denotes the Cartesian product  $\mathcal{V} \times \dots \times \mathcal{V}$  ( $\kappa$  times). By  $X:\mathcal{V} \rightarrow \mathcal{W}$ , we denote a linear transformation operating from  $\mathcal{V}$  to  $\mathcal{W}$ . As usually,  $\text{Im } X = X\mathcal{V}$  denotes the image of  $X$  and  $\ker X$  is its kernel. Moreover,  $X^{-1}\mathcal{T}$  stands for the inverse image of  $\mathcal{T} \subset \mathcal{W}$ . The special subspaces  $\text{Im } B$  and  $\ker C$  are denoted by  $\mathcal{B}$ , and  $\mathcal{K}$ , respectively. The zero dimension subspace is indicated as 0 and  $\sigma\{A\}$  denotes the spectrum of the linear transformation  $A$ . The identity operator is denoted by  $I: Ix = x$ ;  $A^0$  is the identity operator, for any given linear transformation  $A$ . The matrix of a given linear transformation  $A:\mathcal{X} \rightarrow \mathcal{X}$  in a given basis is noted as  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Additionally, for the elements  $v, w, \dots \in \mathcal{X}$  in the given basis, we use the notation  $\mathbf{v}, \mathbf{w}, \dots \in \mathbb{R}^n$ . In our paper,  $\mathbf{1}$  stands for the vector where all its components are equal to 1 and  $\text{BD}\{\mathbf{X}_1, \dots, \mathbf{X}_k\}$  denotes a block diagonal matrix.

$\mathcal{C}_{(M,S)}$  denotes the controllability map of the pair  $(M, S)$ ,  $M:\mathcal{X} \rightarrow \mathcal{X}$  and  $S:\mathcal{Q} \rightarrow \mathcal{X}$ , namely:

$$\mathcal{C}_{(M,S)} = [S \quad MS \quad \dots \quad M^{(n-1)}S]$$

and we denote:  $\langle A | \mathcal{B} \rangle = \text{Im } \mathcal{C}_{(A,B)}$ . We also introduce:

$$\mathcal{C}_{(M,S)}^{[n]} = [S \quad MS \quad \dots \quad M^{(n-1)}S].$$

The unobservable subspace of the pair  $(C, M)$ ,  $C:\mathcal{X} \rightarrow \mathcal{Y}$  and  $M:\mathcal{X} \rightarrow \mathcal{X}$  is denoted by  $\langle \mathcal{K} | M \rangle$ .

- $d^0/dt^0$  stands for the identity operator:  $d^0v/dt^0 = v$ .  $\Psi_n(d/dt)$  stands for the differential operator (from  $\mathcal{Q}$  to  $\mathcal{Q}^n$ ):

$$\Psi_n(d/dt) = [I \quad I d/dt \quad \dots \quad I d^{n-1}/dt^{n-1}]^T.$$

- Given a vector  $\mathbf{x} \in \mathbb{R}^n$ , the expression  $\|\mathbf{x}\|$  means the Euclidean norm and  $\mathcal{B}_\rho$  stands for a specific neighborhood with a radius  $\rho$ :

$$\mathcal{B}_\rho = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < \rho\}.$$

Similarly to the above notation,  $\|\mathbf{x}\|_\infty$  is the usual *sup*-norm and  $C^\infty$  denotes the set of infinitely differentiable functions.

- We next define the matrix  $A_{c_n} \in \mathbb{R}^{n \times n}$ , and the vectors  $\chi_n^i \in \mathbb{R}^{n \times 1}$ ,  $i \in \{1, \dots, n\}$ , as follows:

$$A_{c_n} \triangleq \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \chi_n^1 \triangleq \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \chi_n^2 \triangleq \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, \chi_n^n \triangleq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (1)$$

### 1. Introduction and motivation

Linearization techniques have been established for a long-time as a powerful tool for simplified analysis and design of general nonlinear dynamic systems. “Linearize” the initial nonlinear problem has various meanings in Applied Mathematics, depending on the areas where it is defined, depending also on what one linearizes (a functional, an equation *etc.*). In the context of a robust control design, when dealing with the practical stabilization of the

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx

3

initially given nonlinear dynamic model, the most common way of looking at linearization is to consider a suitable version of the so called feedback linearization [21]. The initial concept of feedback linearization was introduced by Isidori [21] and Slotine and Li [35]. The linearization techniques constitute nowadays a crucial tool of the modern Control Theory and are widely used in various engineering applications.

Recall that a general idea for linearization based on the choice of a different state representation has been established since the work of [21] and [29]. This pioneering technique is in some sense similar to a specific choice of references or suitable coordinate systems in Lagrange mechanics. Roughly speaking, the conventional feedback linearization constitutes an equivalent transformation of the original system models into equivalent dynamic models. This resulting model is a simpler one and possesses some useful behavioral properties. In modern engineering, there are many applications of the various feedback linearization approaches. These applications are devoted (but not restricted) to the helicopters flight dynamics, high-performance aircraft control, industrial robots behavior design, biomedical devices and intelligent vehicles control [14,25,26,30,31,38,42].

Also notice that the main idea of the classic feedback linearization is associated in fact with an algebraic transformation of the initially given nonlinear system (see e.g. [20,21,28,35]). Evidently, a general linearization benefits the engineering analysis and design from the possibility to apply the well-established techniques of the Linear Control Theory. Consider the following (nonlinear) control-affine system:

$$\begin{aligned} \frac{d}{dt}x(t) &= f(x(t)) + g(x(t))u(t), \\ y(t) &= Cx(t), \end{aligned} \tag{1.1}$$

where  $x \in \mathbb{R}^n$  denotes the state variable,  $y \in \mathbb{R}^p$  is an output and  $u \in \mathbb{R}^m$  is the control input. Here  $p \leq m$ ,  $f(x)$  and  $g(x)$  are sufficiently smooth vector fields on  $\mathbb{R}^n$  and  $C \in \mathbb{R}^{p \times n}$ . We now make some technical assumptions that guarantee existence and uniqueness of a local solution to the initial system (1.1).

**H1.** The origin  $x_o = 0 \in \mathbb{R}^n$  is an equilibrium point of the autonomous system (1.1) ( $u = 0$ ). That means  $f(0) = 0$ .

**H2.** The given functions  $f$  and  $g$  are often enough continuously differentiable in a neighborhood of the equilibrium point 0.

From the above assumptions also follows the existence of positive constants  $r_\ell$ ,  $K_{\mathcal{L}_f}$  and  $K_{\mathcal{L}_g}$  such that  $\|f(x_1) - f(x_2)\| \leq K_{\mathcal{L}_f} \|x_1 - x_2\|$ , and  $\|g(x_1) - g(x_2)\| \leq K_{\mathcal{L}_g} \|x_1 - x_2\|$  for all  $x_1, x_2 \in \mathcal{B}_{r_\ell}$ . Note that from H1 also follows the existence of some positive constants  $h_f$  and  $h_g$  such that  $\|f(0)\| \leq h_f$  and  $\|g(0)\| \leq h_g$  for all  $t \geq 0$ . We refer to [39] for the necessary proofs and corresponding analytical details.

In order to have a wide panorama of this paper, a short overview related to some existing (classic) linearization techniques is now given, as well as the principal ideas issued in this contribution. For this, an academic example is introduced, which will be used all along this paper.

**Example 1.** (Academic Example) The PVTOL (Planar Vertical Take off Landing) System of Fig. 1. The PVTOL is one to the most simple (theoretical) aircraft vehicles, which is very close to a (real) Quadrotor Laboratory Prototype. Here  $m_T$  is the total mass,  $J$  is the moment of inertia with respect to the rotation axis  $oy$  and  $L$  denotes the distance from the center of

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

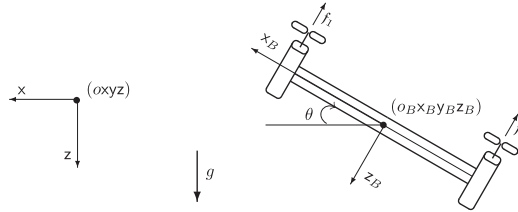


Fig. 1. Schematic PVTOL diagram.

gravity to the thrusters. The motion is considered with respect to a fixed orthogonal axis set  $(oxyz)$ , where  $oz$  points vertically down along the gravity vector  $[0 \ 0 \ g]^T$ . The origin  $o$  is located at the desired height  $\bar{z}$  to the ground level. Moreover,  $\theta$  denotes here the Euler angle determined by the axis  $oByB$ . Here  $(oBxBYBzB)$  is the body axis system with its origin in  $o_B$  (fixed at the centre of gravity of the PVTOL). The PVTOL moves in the plane  $(oxz)$ . By  $f_1, f_2 \in [0, mg)$  we denote the actuators (thrusters).

The mathematical model of the PVTOL system is represented by the following (behavioral) equations. We refer to [11,15] for the analytical details.

$$\begin{bmatrix} d^2x/dt^2 \\ d^2\theta/dt^2 \\ d^2z/dt^2 \end{bmatrix} = \begin{bmatrix} (1/m_T)u_z \tan(\theta) \\ (L/J)u_x \\ (1/m_T)u_z + g \end{bmatrix}. \tag{1.2}$$

Here:

$$u_z = -(f_1 + f_2) \cos \theta \quad \text{and} \quad u_x = (f_1 - f_2).$$

The output and the state variables of the above model can be interpreted as follows:

$$y \doteq x, \\ x \doteq [x \quad dx/dt \quad \theta \quad d\theta/dt \quad z \quad dz/dt]^T.$$

Applying the conventional gravity compensation control law:

$$u_z = m_T([0 \ 0 \ 0 \ 0 \ -a_{z,2} \ -a_{z,1}]x - g) \quad \text{and} \quad u \doteq (L/J)u_x, \tag{1.3}$$

the z-dynamics is represented by the differential equation:

$$(d^2/dt^2 + a_{z,1}d/dt + a_{z,2})z(t) = 0,$$

where  $\pi_z(s) = s^2 + a_{z,1}s + a_{z,2}$  is a Hurwitz polynomial. Finally we deduce the nonlinear state description (cf. Eq. (1.1))

$$\frac{d}{dt}x = \underbrace{\begin{bmatrix} x_2 \\ -(g + a_{z,2}x_5 + a_{z,1}x_6) \tan(x_3) \\ x_4 \\ 0 \\ x_6 \\ -(a_{z,2}x_5 + a_{z,1}x_6) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{g(x)} u, \\ y = \underbrace{[1 \ 0 \ 0 \ 0 \ 0 \ 0]}_{h(x)} x. \tag{1.4}$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>



ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx

5

1.1. An approximated input–output linearization

One of the classic linearization techniques (widely used by “pioneers” of the modern Control Theory) consists in representation of Eq. (1.1) in the form of Taylor’s series around the equilibrium point  $x_o = 0 \in \mathbb{R}^n$  and in the corresponding approximation using the first order terms. Applying this simple idea we get:

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + Bu(t) + f_o(x(t), u(t)), \\ y(t) &= Cx(t), \end{aligned} \tag{1.5}$$

where

$$A \doteq [\partial f / \partial x]_{x=0} \in \mathbb{R}^{n \times n} \quad B \doteq [g]_{x=0} \in \mathbb{R}^{n \times m}, \tag{1.6}$$

and  $x \in \mathcal{X} \approx \mathbb{R}^n$  is the state,  $u \in \mathcal{U} \approx \mathbb{R}^m$  is the input,  $y \in \mathcal{Y} \approx \mathbb{R}^p$  is the output. Moreover, we have:

$$\begin{aligned} f_o(x(t), u(t)) &= \Delta f(x(t)) + \Delta g(x(t))u(t), \\ \Delta f(x(t)) &= f(x(t)) - Ax(t), \\ \Delta g(x(t)) &= g(x(t)) - B. \end{aligned} \tag{1.7}$$

From the above formalism we can deduce that for a sufficiently small neighborhood  $\mathcal{B}_\rho$ :  $|\Delta f(x(t))| \rightarrow 0$  and  $|\Delta g| \rightarrow 0$ , the resulting obtained linear state approximation (for all  $x_\ell \in \mathcal{B}_\rho$ ) can then be written as follows:

$$\begin{aligned} \frac{d}{dt}x_\ell(t) &= Ax_\ell(t) + Bu(t), \\ y(t) &= Cx_\ell(t), \end{aligned} \tag{1.8}$$

This classic linearization technique is a formal consequence of the celebrated Stability Principle of the First Approximation (SPFA). Recall that this principle states that the existence of a state feedback,  $F_{st} \in \mathbb{R}^n$ , which stabilizes the linear state approximation (1.8), implies the required stability property of Eq. (1.5) (by  $F_{st}$ ) in the neighborhood  $\mathcal{B}_\rho$ . We refer to [22,23,33,35,39] for the corresponding technical details.

For the academic example (1.4), the matrices  $A$ ,  $B$  and  $C$  of Eq. (1.8) are (cf. Eq. (1.6)):

$$A = \begin{bmatrix} A_{0,x} & 0 \\ 0 & A_z \end{bmatrix}, \quad B = \begin{bmatrix} B_x \\ 0 \end{bmatrix}, \quad C = [C_x \mid 0], \tag{1.9}$$

where:

$$A_{0,x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{cases} A_z = \bar{A}_z + B_z F_z, F_z = [-a_{z,2} & -a_{z,1}], \\ B_x = \chi_4^4, C_x = (\chi_4^1)^T, \bar{A}_z = A_{e_2}, B_z = \chi_2^2. \end{cases} \tag{1.10}$$

Moreover, we have  $x = [x_x \quad x_z]^T$ ,  $\dot{x}_x = [x \quad dx/dt \quad \theta \quad d\theta/dt]^T$ ,  $x_z = [z \quad dz/dt]^T$ .

We next apply the stabilizing state feedback:

$$u = F_x x_x + \bar{u}_x, \quad \text{where: } F_x = [a_{x,4}/g \quad a_{x,3}/g \quad -a_{x,2} \quad -a_{x,1}]. \tag{1.11}$$

Note that the characteristic polynomial of the closed loop matrix,

$$A_x = A_{0,x} + B_x F_x, \tag{1.12}$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

is the Hurwitz polynomial:

$$\pi_{x_x}(s) = \det(sI - A_x) = s^4 + a_{x,1}s^3 + a_{x,2}s^2 + a_{x,3}s + a_{x,4}. \quad (1.13)$$

Note that the feedback-type control (1.11) has a very simple additive structure. The component  $\bar{u}_x$  in Eq. (1.11) can be directly obtained (using the given forces) from the original systems dynamics (1.2). Moreover, the “gain matrix”  $F_x$  in the above formulae can be directly produced using the elements of the nonlinear state description (1.4). Evidently the proposed control design has a more simple structure in comparison with the conventional PID control law (see e.g., [23]). The same is also true with respect to a possible application of the sliding mode controller or the spline based control techniques [8,23].

The state space description of the obtained closed loop system can be expressed as:

$$\begin{aligned} \frac{d}{dt}x &= \begin{bmatrix} A_x & 0 \\ 0 & A_z \end{bmatrix} x + \begin{bmatrix} B_x \\ 0 \end{bmatrix} \bar{u}_x + \begin{bmatrix} S_x \\ 0 \end{bmatrix} q_x(x), \\ x &= \begin{bmatrix} C_x & 0 \end{bmatrix} x. \end{aligned} \quad (1.14)$$

Here:

$$\begin{aligned} S_x &= \chi_4^2, \\ q_x(x) &= -(a_{z,2}x_5 + a_{z,1}x_6) \tan(x_3) + g(x_3 - \tan(x_3)) = d^2x/dt^2 + g\theta. \end{aligned} \quad (1.15)$$

### 1.2. An exact input–output linearization

The next linearization technique we examine here is the so called “exact input output linearization”. Consider the basic hypotheses H1, H2 and additionally make the following non-restrictive assumption.

**H3.** The dynamic system under consideration has relative degree<sup>1</sup>  $n$  at any point in  $\mathbb{R}^n$

Taking into consideration the above assumptions, the conventional feedback linearization can effectively be applied to the initial system (1.1) (see e.g. [21,23,35]). Let us briefly examine the input–output linearization technique for the specific case:

$$p = m = 1.$$

From H3 it follows that there exists a specific diffeomorphism  $T_{dif} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $\xi(t) = T_{dif}(x(t))$  such that:

$$\begin{aligned} \frac{d}{dt}\xi(t) &= A_{c_n}\xi(t) + \chi_n^n(\alpha(x) + \beta(x)u(t)), \\ y(t) &= (\chi_n^1)^T \xi(t). \end{aligned} \quad (1.16)$$

Here  $\xi := [\xi_1 \ \dots \ \xi_n]^T$ .

To get an explicit linearization control law, the additional hypothesis is added.

**H4.** There exists a known positive constant  $\bar{\beta}$  such that for all  $x$  we have  $0 < \bar{\beta} < \beta(x)$ .

<sup>1</sup> For the definition of relative degree see [21] and [23].

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

*M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx* 7

Taking into consideration the complete knowledge of the system's parameters and applying the ideal control law:

$$u(t) = \beta^{-1}(x)(-\alpha(x) + F_n \xi(t)), \quad F_n = [-a_n \quad \cdots \quad -a_1] \quad (1.17)$$

we obtain the resulting linear closed loop system

$$\begin{aligned} \frac{d}{dt} \xi(t) &= (A_{c_n} + \chi_n^n F_n) \xi(t), \\ y(t) &= (\chi_n^1)^T \xi(t). \end{aligned} \quad (1.18)$$

Here  $\pi_n(s) = s^n + a_1 s^{n-1} + \cdots + a_n$  is a Hurwitz polynomial.

We now apply the exact input-output linearization technique to the academic example (1.4). Differentiating the output and taking into consideration the corresponding relative degree, we observe that:

$$\begin{aligned} \frac{d}{dt} \xi &= \begin{bmatrix} A_{c_4} & 0 \\ 0 & \bar{A}_z \end{bmatrix} \xi + \begin{bmatrix} B_x \\ 0 \end{bmatrix} (\alpha(x) + \beta(x)u), \\ y &= [C_x \quad 0] \xi. \end{aligned} \quad (1.19)$$

We use here the additional notation:

$$\xi = [\xi_x^T \quad x_z^T]^T, \quad \xi_x = [x \quad dx/dt \quad d^2x/dt^2 \quad d^3x/dt^3]^T,$$

$$\begin{aligned} d^2x/dt^2 &= L_f^2(h) = -(g + a_{z,2}z + a_{z,1}dz/dt) \tan(\theta), \\ d^3x/dt^3 &= L_f^3(h) = -d\theta/dt (g + a_{z,2}z + a_{z,1}dz/dt) \sec^2(\theta) \\ &\quad + (a_{z,1}a_{z,2}z + (a_{z,1}^2 - a_{z,2})dz/dt) \tan(\theta), \end{aligned} \quad (1.20)$$

$$\begin{aligned} \alpha(x) &= L_f^4(h) = d\theta/dt (-2 d\theta/dt (g + a_{z,2}z + a_{z,1}dz/dt) \tan(\theta) + a_{z,1}a_{z,2}z \\ &\quad + (a_{z,1}^2 - a_{z,2}) dz/dt) \sec^2(\theta) + a_{z,2} dz/dt (-d\theta/dt \sec^2(\theta) + a_{z,1} \tan(\theta)) \\ &\quad - (a_{z,2}z + a_{z,1}dz/dt) (-a_{z,1}d\theta/dt \sec^2(\theta) + (a_{z,1}^2 - a_{z,2}) \tan(\theta)), \\ \beta(x) &= L_g^1(L_f^3(h)) = -(g + a_{z,2}z + a_{z,1}dz/dt) \sec^2(\theta). \end{aligned} \quad (1.21)$$

Implementing the stabilizing state feedback (cf. Eq. (1.17))  $u = \beta^{-1}(x)(-\alpha(x) + F_4 \xi + \bar{u}_x)$ ,  $F_4 = [-a_{x,4} \quad -a_{x,3} \quad -a_{x,2} \quad -a_{x,1}]$ , we finally get the closed loop state space description:

$$\begin{aligned} \frac{d}{dt} \xi &= \begin{bmatrix} A_{c_4 F_4} & 0 \\ 0 & \bar{A}_z \end{bmatrix} \xi + \begin{bmatrix} B_x \\ 0 \end{bmatrix} \bar{u}_x, \\ \times &= [C_x \quad 0] \xi. \end{aligned} \quad (1.22)$$

Note that the characteristic polynomial of the closed loop matrix  $A_{c_4 F_4} = A_{c_4} + B_x F_4$  is the Hurwitz polynomial (1.13).

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

### 1.3. Exact structural linearization

With the help of the academic example (1.4), we now introduce an exact linearization technique based on the system structure. For this, we first apply the change of variable:<sup>2</sup>

$$\zeta_x = x_x - \frac{1}{g} \begin{bmatrix} 0 \\ 0 \\ 1 \\ d/dt \end{bmatrix} q_x(x) \quad (1.23)$$

to the closed loop system described by Eq. (1.14). We next get the state space realization:

$$\begin{aligned} \frac{d}{dt} \zeta &= \begin{bmatrix} A_x & 0 \\ 0 & A_z \end{bmatrix} \zeta + \begin{bmatrix} B_x \\ 0 \end{bmatrix} (\bar{u}_x + q_{x,*}(x)), \\ x &= [C_x \mid 0] \zeta. \end{aligned} \quad (1.24)$$

Observe that  $\zeta = [\zeta_x \quad x_z]^T$  and:<sup>3</sup>

$$\begin{aligned} q_{x,*}(x) &= -(1/g)(d^2/dt^2 + a_{x,1} d/dt + a_{x,2})q_x(x) \\ &= -(a_{x,1}/g)(d^3x/dt^3 + g d\theta/dt) - (a_{x,2}/g)(d^2x/dt^2 + g\theta) \\ &\quad - (1/g)(\alpha(x) + (\beta(x) + g) d^2\theta/dt^2). \end{aligned} \quad (1.25)$$

We also get an expression for the transfer function of the state space realization (1.24):<sup>4</sup>

$$F_{\zeta_x}(s) = C_x(sI - A_x)^{-1}B_x = -g/\pi_{x_x}(s). \quad (1.26)$$

We can now conclude that with the following control law:

$$\bar{u}_x = -q_{x,*}(x)$$

system (1.24) is exactly linearized.

### 1.4. Robust asymptotic feedback linearization

If the computation of Eq. (1.25) would not be possible or would be a hard work, we could try to estimate it. Based on the Beard–Jones filter (cf. [4,19,27,36,41]), the following dynamic output feedback is proposed:<sup>5</sup>

$$\begin{aligned} \frac{d}{dt} w_x &= (A_{K_x} + B_x G_x^\ell C_x) w_x - (K_x + B_x G_x^\ell) x, \\ \bar{u}_x &= G_x^\ell C_x w_x - G_x^\ell x, \end{aligned} \quad (1.27)$$

where:

<sup>2</sup> In Section 2.2, we formalize this change of variable, and the computations are done in Example 2.

<sup>3</sup> Recall Eqs. (1.14), (1.15), (1.12), (1.10), (1.11), and:  $d^4x/dt^4 = \alpha(x) + \beta(x) d^2\theta/dt^2$  (see Eqs. (1.19) and (1.4)).

<sup>4</sup> Recall Eqs. (1.12), (1.10), (1.11) and (1.13).

<sup>5</sup> This control law is studied in Section 3, and the computations are done in Example 3.

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx 9

$$A_{K_x} = A_x + K_x C_x, \quad G_x^\ell = -(C_x A_{K_x}^{-1} B_x)^\ell,$$

$$K_x = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{x,4} - a_{x_0,4} \\ a_{x,3} - a_{x_0,3} \\ a_{x,2} - a_{x_0,2} \\ a_{x,1} - a_{x_0,1} \end{bmatrix}. \quad (1.28)$$

The characteristic polynomial of  $A_{K_x}$  is the Hurwitz polynomial:

$$\pi_{e_x}(s) = \det(sI - A_{K_x}) = s^4 + a_{x_0,1} s^3 + a_{x_0,2} s^2 + a_{x_0,3} s + a_{x_0,4}. \quad (1.29)$$

Introducing the new state variable,  $e_x = w_x - \zeta_x$ , we get from Eqs. (1.27) and (1.24), the modified closed loop state description:

$$\frac{d}{dt} \begin{bmatrix} e_x \\ \zeta_x \\ x_z \end{bmatrix} = \underbrace{\begin{bmatrix} A_{K_x} & 0 & 0 \\ B_x G_x^\ell C_x & A_x & 0 \\ 0 & 0 & A_z \end{bmatrix}}_{A_{CL}} \begin{bmatrix} e_x \\ \zeta_x \\ x_z \end{bmatrix} + \underbrace{\begin{bmatrix} -B_x \\ B_x \\ 0 \end{bmatrix}}_{B_{CL}} q_{x,*}(x),$$

$$y = \underbrace{\begin{bmatrix} 0 & C_x & 0 \end{bmatrix}}_{C_{CL}} \begin{bmatrix} e_x \\ \zeta_x \\ x_z \end{bmatrix}. \quad (1.30)$$

The transfer functions involved in the state space realization (1.30) are defined here:<sup>6</sup>

$$F_{e_x}(s) = G_x^\ell C_x (sI - A_{K_x})^{-1} B_x = a_{x_0,4} / \pi_{e_x}(s), \quad (1.31)$$

$$F_{CL}(s) = C_{CL} (sI - A_{CL})^{-1} B_{CL}$$

$$= \begin{bmatrix} 0 & C_x \end{bmatrix} \begin{bmatrix} (sI - A_{K_x})^{-1} & 0 \\ (sI - A_x)^{-1} B_x G_x^\ell C_x (sI - A_{K_x})^{-1} & (sI - A_x)^{-1} \end{bmatrix} \begin{bmatrix} -I \\ I \end{bmatrix} B_x$$

$$F_{CL}(s) = F_{\zeta_x}(s) (1 - F_{e_x}(s)) = -\frac{g}{\pi_{x_x}(s)} \left( 1 - \frac{a_{x_0,4}}{\pi_{e_x}(s)} \right) = -g \frac{s \bar{\pi}_{w_x}(s)}{\pi_{x_x}(s) \pi_{e_x}(s)}. \quad (1.32)$$

We use above the additional notation  $\bar{\pi}_{w_x}(s) = s^3 + a_{x_0,1} s^2 + a_{x_0,2} s + a_{x_0,3}$ .

### 1.5. Discussion

Let us compare the linearization techniques overviewed in Section 1.

1. *The approximated input–output linearization:* A natural approach to control nonlinear systems is to use the classic Taylor approximation. This is in fact a very simple approach. However, the corresponding techniques must be applicable in a small neighborhood of a fixed set point. Under some restrictive assumptions related to the external disturbance, this technique can be useful for classes of regulation problems. On the other hand, it is not recommended, for example, in tracking control problems.

2. *Exact input–output linearization:* The feedback linearization problem is implemented by a (specific) exact state transformation and an application of the feedback control strategy. Notice that this analytic approach requires a complete knowledge of the dynamic model

<sup>6</sup> Recall Eqs. (1.28), (1.26), (1.29), (1.10), (1.11) and (1.12).

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

parameters as well as of the corresponding derivatives. However, in many engineering applications, this assumption is not valid. This is true, for example, for problems from the area of industrial robotics, helicopters control, high-performance aircraft control. This situation also occurs in biomedicine and vehicles control (see [37]). To study this problem, many alternative solution approaches have been proposed. We refer to [1,3,8,10,13,18,24] for some well-established and practically implementable feedback linearization techniques in the absence of the exact systems description. In all the works mentioned above, it is required to make some (usually restrictive) additional assumptions related to the main system model. We can mention the so called “communication” system or external dynamics or the complete state measurability assumptions. Evidently, for the general nonlinear systems and also for the control-affine class, these *a priori* assumptions are difficult to verify.

3. *Exact structural linearization:* In Section 1.3, we propose an exact linearization methodology based on a change of variable (cf. Eq. (1.23)). In Theorem 1 (see Section 2), we explain in details how to find it. Additionally in Lemma 1, we show that this change of variable is supported by the controllability structure of the linearized system (obtained by means of the first order Taylor series based approximation).

Let us compare the exact structural feedback linearization introduced in Section 1.3 and the well known exact input–output linearization, recalled in Section 1.2. In order to enlighten the main idea, consider here  $a_{x,4}=a_{x,3}=a_{x,2}=a_{x,1}=0$ . From Eqs. (1.11), (1.4) and taking into consideration (1.25), we next get the specific control design that corresponds to the proposed linearization

$$u = \bar{u}_x, \quad d^2\theta/dt^2 = u \quad \text{and} \quad q_{x,*}(x) = -(1/g)(\alpha(x) + (\beta(x) + g)u).$$

Additionally, we also obtain the useful relations (recall Eqs. (1.24), (1.23) and (1.15)):

$$\begin{aligned} (\alpha(x) + \beta(x)u) &= -g(u + q_{x,*}(x)) = -g[0 \quad 0 \quad 0 \quad 1]^T d\zeta_x/dt \\ &= -gd^2\theta/dt^2 + d^2q_{x,*}/dt^2 \\ (\alpha(x) + \beta(x)u) &= d^4x/dt^4, \end{aligned}$$

which can be exactly identified with Eq. (1.19).

Note that the above compliance is not a surprising result. Both of the considered procedures are in fact exact linearizations and naturally lead to the same formal result. The formal difference consists in the manner of obtaining this exact linearization. The advantage of the linear representation which we propose is that the nonlinear uncertainty signal  $q_*$  can be treated as an additive disturbance signal. Note that this approach constitutes a specific MIMO procedure that pre-stabilizes the given system matrix  $A$ .

4. *Robust asymptotic feedback linearization:* Some important features of the indicated control scheme are emphasized hereafter.

1. Since the nonlinear uncertainty signal  $q_*$  obtained in Section 1.3 can be treated as an additive disturbance signal (cf. Eq. (1.24)), one can then use the well developed linear tools for the disturbances rejections (as for example the Beard–Jones filter (1.27)).
2. For synthesizing the Beard–Jones filter (1.27) the only information one needs is the linear approximation of first variation (1.8) (Eqs. (1.9)–(1.12) in example).
3. The transfer function (1.32) of the closed-loop state space representation (1.30) constitutes a typical low-pass filter. If the norm of the time derivative of the nonlinear uncertainty signal (1.25) is bounded  $\|dq_{x,*}/dt\|_\infty \leq \mathcal{S}_{\mathcal{R}}$  ( $q_{x,*}$  has a finite slew rate  $\mathcal{S}_{\mathcal{R}}$ ),

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx 11

this implies that the steady-state response of Eq. (1.30) can be attenuated by increasing the bandwidth of the Beard–Jones filter (1.27). This fact is formally proved in our Lemmas 2 and 3 of Section 3.

4. From consideration of the transfer function (1.31), we can deduce that the remainder generator  $\hat{q}_{x,*} = -G_x^\ell C_x e$  of the Beard–Jones filter (1.27) is in fact a reconstructor of the nonlinear uncertainty signal  $q_{x,*}$ . Therefore the “increment” of the bandwidth of the Beard–Jones filter (1.27) implies that the remainder generator  $\hat{q}_{x,*}$  tends to the nonlinear uncertainty signal  $q_{x,*}$ . The last one implies its attenuation (see the algebraic part of Eq. (1.27)). Resulting from that, the usual neighborhood stability will be extended (in the sense of an extended neighborhood). This useful property of the proposed control design is formally proved in Corollary 1 (see Section 4).

When using linearization techniques in control, one needs to discuss the conservativeness of the resulting control design approach. This necessity is a simple consequence of the obvious approximating character of the linearization procedures with respect to the initially given dynamics. We discuss these conservativeness [16,17,23] aspects after our main theoretical results, namely, after Theorem 2 and Corollary 1, in Section 4.

The remainder of our paper is organized as follows.

In Section 2, we present our main result, namely, the structural feedback linearization technique that we propose. The robust asymptotic extension of the resulting structural feedback linearization is discussed in Section 3. Section 4 is devoted to some theoretical extensions of the main linearization idea proposed in Section 2. The practically oriented example of a PVTOL system is used throughout the paper to illustrate the efficiency of the proposed linearization approach. This culminates in Section 5 that is devoted to the numerical treatment of the PVTOL control. Section 6 summarizes our paper.

## 2. The structural linearization approach

This section introduces a novel linearization technique for the class of control-affine dynamic systems under consideration. The proposed linearization methodology is based on the given internal structure of the state description of the linearized system characterized by the pair  $(A, B)$ , with:

$$A \doteq [\partial f / \partial x]_{x=0} \quad \text{and} \quad B \doteq [g]_{x=0}.$$

For this reason, we characterize formally the admissible system nonlinearities (see Theorem 1) which are specified by an additional nonlinear perturbation signal  $q$ . This characterization is based on a suitable change of variable obtained by means of an auxiliary algebraic structural equation, stated in Lemma 1. The last result makes it possible to map the nonlinear perturbation signal  $q$  into a new *nonlinear perturbation signal*  $q_*$ .

Let us now introduce some additional technical assumptions.

### H5. Structural assumptions:

1.  $B$  is monic, that is:  $\ker B = 0$ .
2. The pair  $(A, B)$  is controllable.
3. The pair  $(C, A)$  is observable.

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

For the technical proofs of our main theoretic results, we also need some auxiliary results and facts that are collected in the following subsection.

### 2.1. The structural decomposition

Consider the following assumption.

**H6.** There exist an uncertainty vector (nonlinear perturbation signal)  $q \in \mathcal{Q} \approx \mathbb{R}^\mu$  and a linear transformation  $S : \mathcal{Q} \rightarrow \mathcal{X}$  such that:

$$f_o(x(t), u(t)) = S q(x(t), u(t)). \quad (2.1)$$

Application of the above transformation to the initially given system (1.5), implies the rewritten dynamic model:

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + Bu(t) + S q, \\ y(t) &= Cx(t). \end{aligned} \quad (2.2)$$

We now prove our next result.

**Lemma 1.** Assume that H5.1 and H5.2 are true. Then:

1. the images of  $A$  and  $B$  span the whole state space  $\mathcal{X}$ :

$$\text{Im } A + \mathcal{B} = \mathcal{X}; \quad (2.3)$$

2. the linear transformation  $[A \ B] : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$  is right invertible and there exist linear transformations  $M : \mathcal{X} \rightarrow \mathcal{X}$  and  $X : \mathcal{X} \rightarrow \mathcal{U}$  such that:

$$AM + BX = I; \quad (2.4)$$

3. there exists a nilpotent solution  $M : \mathcal{X} \rightarrow \mathcal{X}$  of Eq. (2.4) which satisfies:

$$M^n = 0. \quad (2.5)$$

**Proof.** From H5.2, we get:  $\mathcal{X} = (A | \mathcal{B}) \subset \text{Im } A + \mathcal{B} \subset \mathcal{X}$ , which implies:  $\text{Im } A + \mathcal{B} = \mathcal{X}$ . The right invertibility (2.4) of  $[A \ B]$  is a direct consequence of Eq. (2.3).

From Eq. (2.4), we conclude that  $\text{Im } (I - AM) \subset \text{Im } B = \ker \widehat{N}$ , where  $\widehat{N} : \mathcal{X} \rightarrow \mathcal{X} / \mathcal{B}$  is the canonical projection, thus:

$$\widehat{N}AM = \widehat{N}. \quad (2.6)$$

From H5.1, there exists a left inverse  $B^\ell : \mathcal{X} \rightarrow \mathcal{U}$  of  $B : \mathcal{U} \rightarrow \mathcal{X}$ , then once found a matrix  $M$  solving Eq. (2.6), one solution of Eq. (2.4) is:

$$X = B^\ell (I - AM), \quad (2.7)$$

Using now the celebrated Brunovsky's Theorem [9], one can deduce the existence of bases in  $\mathcal{X}$ ,  $\mathcal{U}$  and  $\mathcal{X} / \mathcal{B}$  with the property:

$$\widehat{N}A = \text{BD}\{G_1, \dots, G_m\} \quad \text{and} \quad \widehat{N} = \text{BD}\{N_1, \dots, N_m\}, \quad (2.8)$$

Here  $G_i$  and  $N_i$  are matrices in  $\mathbb{R}^{(\kappa_{i-1}) \times \kappa_i}$ ,  $i \in \{1, \dots, m\}$  with the property  $\sum_{i=1}^m \kappa_i = n$  and:

$$G_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad \text{and} \quad N_i = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}. \quad (2.9)$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>



ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx 13

The last relation implies that the matrix  $M$  is defined as follows<sup>7</sup>:

$$M = \text{BD}\{M_{1_{\kappa_1}}, \dots, M_{m_{\kappa_m}}\}, M_{i_{\kappa_i}} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{\kappa_i \times \kappa_i}, \quad (2.10)$$

where  $i \in \{1, \dots, m\}$ . It satisfies conditions (2.4) and (2.5). The proof is completed.  $\square$

We are now ready to present our first main result. For a better readability, this result is organized in a separated subsection.

2.2. The exact scheme of the structural feedback Linearization

The next technical assumption constitutes in fact a future restriction of the dynamic models under consideration. On the other hand, novel control design techniques (in our case a linearization based design) are nowadays associated with some subclasses of general control systems. These particular subclasses are of a specific interest in some areas of the modern Control Engineering and one can treat and “robustify” these systems using some specific properties of the given dynamics. Exactly these properties constitute in fact the above mentioned systems restrictions.

**H7.** The subspace  $M \text{Im } S$  is contained in the unobservable subspace  $\langle \mathcal{X} | M \rangle$ , namely:

$$C M C_{(M,S)} = 0. \quad (2.11)$$

**Theorem 1.** Given the linear transformations  $M$  and  $X$  from Lemma 1, consider the following change of variable:

$$\zeta(t) = x(t) + M C_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)). \quad (2.12)$$

Under Assumptions H5.1, H5.2 and H7, the state representation (2.2) is externally equivalent to the following:<sup>8</sup>

$$\begin{aligned} \frac{d}{dt} \zeta(t) &= A \zeta(t) + B(u(t) + q_*(x(t), u(t))), \\ y(t) &= C \zeta(t), \end{aligned} \quad (2.13)$$

where the nonlinear uncertainty signal  $q_*$  is given by:

$$q_*(x(t), u(t)) = X C_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)). \quad (2.14)$$

**Proof.** From Eqs. (2.12) and (2.11), we deduce:

$$C \zeta(t) = C x(t) + C M C_{(M,S)} \Psi_n(d/dt) q(x(t), u(t)) = C x(t). \quad (2.15)$$

Applying the change of variable Eqs. (2.12)–(2.2), and taking into account (2.4) and (2.5), we get:

<sup>7</sup> Notice that this matrix  $M$  corresponds to the transpose matrix of the Brunovsky canonical form of  $A$ .

<sup>8</sup> Recall that two representations are called externally equivalent if the corresponding sets of all possible trajectories for the external variables expressed in an input/output partition  $(u, y)$  are the same [33,40].

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

$$\begin{aligned}
\frac{d}{dt}\zeta &= A\left(\zeta - M\sum_{i=0}^{n-1}M^iS\frac{d^i}{dt^i}q\right) + Bu + Sq + M\sum_{i=0}^{n-1}M^iS\frac{d^{i+1}}{dt^{i+1}}q \\
&= A\zeta + (BX - I)\sum_{i=0}^{n-1}M^iS\frac{d^i}{dt^i}q + Bu + Sq + \sum_{i=1}^{n-1}M^iS\frac{d^i}{dt^i}q \\
&= A\zeta + B\left(u + X\sum_{i=0}^{n-1}M^iS\frac{d^i}{dt^i}q\right).
\end{aligned}$$

The proof is completed.  $\square$

**Theorem 1** makes it possible to represent an initially given sophisticated system (2.2) in a specific form which includes an uncertainty signal explicitly determined. We can next use this result and easily define a control strategy that constitutes a feedback linearization. Following this linearization idea, we conclude that the input design:

$$u(t) = -q_*(x(t), u(t)), \quad (2.16)$$

for the dynamic system described by Eq. (2.13) represents an exact feedback linearizing control.

Hereafter, some conceptual observations are done.

**Remark 1.** Roughly speaking, condition (2.11)  $M \text{Im} S \subset \langle \mathcal{K} | M \rangle$  implies that  $M \text{Im} S$  is contained in the *output decoupling zeros* modes [2,34] of the Brunovsky dual system (cf. Eq. (2.10))  $d\bar{x}/dt = M\bar{x} + Sq$  and  $\bar{y} = C\bar{x}$ .

**Remark 2.** Condition (2.11)  $M \text{Im} S \subset \langle \mathcal{K} | M \rangle$  involves that any trajectory  $\bar{x}(\cdot)$  of the Brunovsky dual system (cf. Eq. (2.10))  $d\bar{x}/dt = M\bar{x}$  and  $\bar{y} = C\bar{x}$  (such that the initial condition belongs to  $M \text{Im} S$ ) is unobservable.

**Remark 3.** When the pair  $(A, B)$  of the system represented by Eq. (2.2) is controllable, then the algebraic equation (2.4) is solvable, and thus from **Theorem 1**, we get the externally equivalent representation (2.13). Notice that  $q_*$  is contained in  $\text{Im} B$ , so its cancelation is not due to any pole-zero cancelation. This fact remains true for the condition:  $\text{Im} S \subset \langle A | \mathcal{B} \rangle$ . In the case that  $\text{Im} S \not\subset \langle A | \mathcal{B} \rangle$ , but  $\text{Im} S \subset \mathcal{V}^* + \langle A | \mathcal{B} \rangle$ , where  $\mathcal{V}^*$  is the supremal  $(A, B)$ -invariant subspace contained in  $\ker C$ ,  $\sup\{\mathcal{V} \subset \ker C \mid \exists F: (A + BF)\mathcal{V} \subset \mathcal{V}\}$ , one can then apply a friend state feedback,  $F_* \in \{F: \mathcal{X} \rightarrow \mathcal{U} \mid (A + BF)\mathcal{V}^* \subset \mathcal{V}^*\}$ , for getting [45]:  $\Pi \text{Im} S \subset \Pi \langle A + BF_* | \mathcal{B} \rangle$ , where  $\Pi$  is the canonical projection of  $\mathcal{X}$  onto  $\mathcal{X}/\mathcal{V}^*$ . Hence, **Theorem 1** is applied to the system quotiented by  $\mathcal{V}^*$ . In this quotient procedure could occur a pole-zero cancelation, and one must thus check that there is no non-Hurwitz zero cancelation.

The second technical part of our main Example is now presented.

**Example 2** (Structural linearization approach). Consider the Academic **Example 1** introduced in **Section 1** and use the main **Theorem 1**. Taking into account (1.6)–(1.7) in (1.4) and applying the stabilizing state feedback (1.11), we get Eq. (1.14).

Recall that  $\pi_x(s) = \det(sI - A_x)$  is a Hurwitz polynomial (cf. Eq. (1.13)), and notice that (i)  $\ker B_x = 0$ , (ii) the pair  $(A_x, B_x)$  is controllable and (iii) the pair  $(C_x, A_x)$  is observable.

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx

15

We now consider the solution of equation  $A_x M_x + B_x X_x = I_4$  that can formally be expressed as:

$$M_x = T_x M_{I_4} T_x^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1/g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, X_x = B_x^\ell (I_4 - A_x M_x) = \begin{bmatrix} -a_{x,3}/g \\ -a_{x,2}/g \\ a_{x,1} \\ 1 \end{bmatrix}^T. \quad (2.17)$$

From Eqs. (2.17) and (1.12) we next deduce:

$$\begin{aligned} A_x M_x + B_x X_x &= I_4, \quad M_x^4 = 0, \quad C_x M_x C_{(M_x, S_x)} = 0, \\ M_x C_{(M_x, S_x)} \Psi_4(d/dt) &= -(1/g) \begin{bmatrix} 0 & 0 & 1 & d/dt \end{bmatrix}^T, \\ X_x C_{(M_x, S_x)} \Psi_4\left(\frac{d}{dt}\right) &= -(1/g)(d^2/dt^2 + a_{x,1} d/dt + a_{x,2}). \end{aligned} \quad (2.18)$$

Moreover, from Eqs. (2.18), (2.12) and (2.14), it follows that the change of variable  $\zeta_x = x_x + M_x C_{(M_x, S_x)} \Psi_4(d/dt) q_x(x)$  has the form (1.23) and that the nonlinear uncertainty signal  $q_{x,*}(x) = X_x C_{(M_x, S_x)} \Psi_4\left(\frac{d}{dt}\right) q_x(x)$  has the form (1.25).

### 3. Robust asymptotic feedback linearization

Recall that the exact feedback linearization (2.16) usually requires a complete knowledge of the system model. Moreover, one also needs the exact formal expressions of the necessary time derivatives in the given nonlinear perturbation signal  $q(x, u)$  (cf. Eq. (2.14)). On the other hand, the approach that we propose has some evident advantages in comparison with the mentioned classic feedback linearizations. Note that we deal with *a priori* unknown nonlinearities  $q_*(x, u)$  that belong to  $\mathcal{B}$ . In the context of Assumption H5.3 these unknown nonlinearities can effectively be treated with disturbance rejectors based on standard state observers. Let us now introduce the necessary formal hypothesis.

**H8.** The state space description  $\Sigma(A, B, C)$  (2.2) has no finite invariant zeros at the origin. That formally means:

$$\text{Im } B \cap A \ker C = 0. \quad (3.1)$$

Assumptions H5.3 and H8 make it possible to apply results of [5] and to design a robust disturbance rejection (based on the Beard–Jones filter (cf. [4,19,27,36,41])):

$$\begin{aligned} \frac{d}{dt} w(t) &= (A + KC) w(t) - Ky(t) + Bu(t), \\ \hat{q}_*(t) &= -G^\ell (C w(t) - y(t)), \end{aligned} \quad (3.2)$$

and:

$$u(t) = u_c(t) - \hat{q}_*(t). \quad (3.3)$$

Here  $K : \mathcal{Y} \rightarrow \mathcal{U}$  is an “output injection” that can be computed as follows:

$$\sigma\{(A + KC)\} \subset \mathbb{C}^-, \quad (3.4)$$

and  $G^\ell$  is a left inverse of the static gain  $-C(A + KC)^{-1}B$ . The remainder generator is expressed as:

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

$$\begin{aligned} \frac{d}{dt}e(t) &= A_K e(t) - B q_*(x, u), \\ \hat{q}_*(t) &= -G^\ell C e(t), \end{aligned} \tag{3.5}$$

where  $A_K \doteq (A + KC)$ , and  $e(t) = w(t) - \zeta(t)$ .

When the classic Laplace transform of  $q_*(x, u)$  is well-defined, we get:

$$\hat{q}_*(s) = G^\ell F(s) q_*(s). \tag{3.6}$$

Under the natural boundedness assumptions for  $q(x, u)$  with a bandwidth  $\omega_c$ , we have to synthesize a Hurwitz low-pass filter  $F(s)$  with a corner frequency  $\omega_c$ . This is with the aims to achieve a robust disturbance in a neighborhood around the equilibrium point  $(x, u) = (0, 0)$ . We also deduce:

$$\|q_*(\omega) - \hat{q}_*(\omega)\| \leq \|(I - G^\ell F(j\omega))X C_{(M,S)} \Psi_n(j\omega)\| \|q(\omega)\|. \tag{3.7}$$

We now study the third technical part of the main illustrative example.

**Example 3** (Robust asymptotic linearization approach). From Eqs. (1.24), (1.12), (3.2) and (3.3), we deduce the state description (cf. Eq. (1.27)):

$$\begin{aligned} \frac{d}{dt}\bar{w}_x &= (\bar{A}_{\bar{K}_x} + \bar{B}_x G_x^\ell \bar{C}_x)\bar{w}_x - (\bar{K}_x + \bar{B}_x G_x^\ell)x, \\ \bar{u}_x &= G_x^\ell \bar{C}_x \bar{w}_x - G_x^\ell x, \end{aligned} \tag{3.8}$$

where

$$\begin{aligned} \bar{w}_x &= T_{ox}^{-1} w_x, T_{ox} = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1}, \quad \bar{A}_{\bar{K}_x} = \bar{A}_x + \bar{K}_x \bar{C}_x, (\bar{A}_x, \bar{B}_x, \bar{C}_x, \bar{K}_x) \\ &= (T_{ox}^{-1} A_x T_{ox}, T_{ox}^{-1} B_x, C_x T_{ox}, T_{ox}^{-1} K_x), G_x^\ell = -(C_x A_{\bar{K}_x}^{-1} B_x)^\ell = -(\bar{C}_x \bar{A}_{\bar{K}_x}^{-1} \bar{B}_x)^\ell, \end{aligned}$$

and (cf. Eq. (1.28))

$$\begin{aligned} \bar{A}_x &= \begin{bmatrix} 0 & 0 & 0 & -a_{x,4} \\ 1 & 0 & 0 & -a_{x,3} \\ 0 & 1 & 0 & -a_{x,2} \\ 0 & 0 & 1 & -a_{x,1} \end{bmatrix}, \quad \bar{B}_x = \begin{bmatrix} -g \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{K}_x = \begin{bmatrix} a_{x,4} - a_{x,4} \\ a_{x,3} - a_{x,3} \\ a_{x,2} - a_{x,2} \\ a_{x,1} - a_{x,1} \end{bmatrix}, \\ G_x^\ell &= -(\bar{C}_x \bar{A}_{\bar{K}_x}^{-1} \bar{B}_x)^\ell = -a_{x,4}/g, \quad \bar{C}_x = [0 \quad 0 \quad 0 \quad 1]. \end{aligned} \tag{3.9}$$

The characteristic polynomial and the corresponding transfer function are expressed as follows (cf. Eqs. (1.31) and (1.29)):

$$\begin{aligned} \pi_{e_x}(s) &= \det(sI - \bar{A}_{\bar{K}_x}) = s^4 + a_{x,1} s^3 + a_{x,2} s^2 + a_{x,3} s + a_{x,4}, \\ F_{e_x}(s) &= G_x^\ell \bar{C}_x (sI - \bar{A}_{\bar{K}_x})^{-1} \bar{B}_x = a_{x,4}/\pi_{e_x}(s). \end{aligned} \tag{3.10}$$

The characteristic polynomial, the system zeros polynomial and the transfer function of the Beard-Jones filter (3.8) have the known expressions (recall Eq. (1.13)):

$$\pi_{\bar{w}_x}(s) = \det(sI - (\bar{A}_{\bar{K}_x} + \bar{B}_x G_x^\ell \bar{C}_x)) = s \bar{\pi}_{\bar{w}_x}(s),$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx 17

$$\bar{\pi}_{\bar{w}_x}(s) = s^3 + a_{x_o,1} s^2 + a_{x_o,2} s + a_{x_o,3}, \quad (3.11)$$

$$\varphi_{\bar{w}_x}(s) = \det \left[ \begin{array}{c|c} sI - (\bar{A}_{\bar{K}_x} + \bar{B}_x G_x^\ell \bar{C}_x) & -(\bar{K}_x + \bar{B}_x G_x^\ell) \\ \hline -G_x^\ell \bar{C}_x & -G_x^\ell \end{array} \right] = -G_x^\ell \pi_{x_x}(s), \quad (3.12)$$

$$F_{\bar{w}_x}(s) = \varphi_{\bar{w}_x}(s)/\pi_{\bar{w}_x}(s) = -G_x^\ell \pi_{x_x}(s) / (s \bar{\pi}_{\bar{w}_x}(s)). \quad (3.13)$$

Note that:

$$s \bar{\pi}_{\bar{w}_x}(s) + a_{x_o,4} = \pi_{e_x}(s). \quad (3.14)$$

The last equation is of a primary importance because it makes it possible to define the Hurwitz polynomials  $\bar{\pi}_{\bar{w}_x}(s)$  and  $\pi_{e_x}(s)$ . In that case, the transfer function of the closed loop system (3.8) and (1.24) has a self-closed form (cf. Eq. (1.32)):<sup>9</sup>

$$F_{cl}(s) = \frac{F_{\zeta_x}(s)}{1 - F_{\zeta_x}(s)F_{\bar{w}_x}(s)} = \frac{-g}{\pi_{x_x}(s)} \frac{s \bar{\pi}_{\bar{w}_x}(s)}{(s \bar{\pi}_{\bar{w}_x}(s) + a_{x_o,4})} = -g \frac{s \bar{\pi}_{\bar{w}_x}(s)}{\pi_{x_x}(s) \pi_{e_x}(s)}. \quad (3.15)$$

If  $q_{x,*}(x)$  is a bounded signal, its steady-state response tends to zero. If it is a signal with a bounded slew rate, then its steady-state response can be attenuated by increasing the bandwidth  $a_{x_o,4}/a_{x_o,3}$ . These conclusions and the possibility to study the corresponding equivalent representations of the given control-affine dynamic systems are direct consequences of the main Theorem 1.

We now consider dynamic systems (2.13), (3.5) and (3.3). The corresponding closed-loop model is expressed as follows (cf. Eq. (1.30)):

$$\begin{aligned} \frac{d}{dt} \zeta(t) &= A \zeta(t) + B(u_c(t) + \tilde{q}_*(t)), \\ \frac{d}{dt} e(t) &= A_K e(t) - B q_*(x, u), \\ \tilde{q}_*(t) &= q_*(t) + G^\ell C e(t), \\ y(t) &= C \zeta(t). \end{aligned} \quad (3.16)$$

The time response of the uncertainty error  $\tilde{q}_*(t) = q_*(t) - \hat{q}_*(t)$  has the form:

$$\begin{aligned} \tilde{q}_*(t) &= q_*(t) + G^\ell C \left( \exp(A_K t) e(0) - \int_0^t \exp(A_K(t - \tau)) B q_*(\tau) d\tau \right), \\ &= G^\ell C \exp(A_K t) \tilde{k}_0 - G^\ell C A_K^{-1} \int_0^t \exp(A_K(t - \tau)) B \frac{dq_*(\tau)}{d\tau} d\tau, \end{aligned} \quad (3.17)$$

where  $\tilde{k}_0 = e(0) - A_K^{-1} B q_*(0)$ . The time response of the output in that case can be directly calculated:

$$y(t) = C \exp(At) \zeta(0) + C \int_0^t \exp(A(t - \tau)) B (u_c(\tau) + \tilde{q}_*(\tau)) d\tau. \quad (3.18)$$

We additionally assume that the uncertainty signal  $q_*$  has a finite slew rate  $\mathcal{S}_{\mathcal{R}}$ :

<sup>9</sup> Recall that:  $G_x^\ell = -a_{x_o,4}/g$ ; see Eq. (3.9).

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

**H9.**  $\exists \mathcal{S}_{\mathcal{R}} \in \mathbb{R}^+$  and  $\|dq_*/dt\|_{\infty} \leq \mathcal{S}_{\mathcal{R}}$ .

Under the above hypothesis we obtain the next analytic result.

**Lemma 2.** Assume that the spectrum  $\sigma_k$  of  $A_K$ , has no repeated eigenvalues. Denote by  $\bar{\omega}_i$  the modulus of each eigenvalue of  $\sigma_k$  and assume that the elements of  $\sigma_k$  have been re-ordered with  $\bar{\omega}_1 \leq \dots \leq \bar{\omega}_n$ . Moreover, let  $\varrho_i$  be scale factors of the natural frequencies  $\bar{\omega}_i$ , namely  $\varrho_i = \bar{\omega}_i/\bar{\omega}_1$ . There then exist positive constants  $k_1, k_3$  and  $\alpha_0$  such that:

$$\|\tilde{q}_*(t)\| \leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + (k_3/\alpha_0) \mathcal{S}_{\mathcal{R}} \frac{\varrho_n^{n-1}}{\bar{\omega}_1}. \tag{3.19}$$

Moreover, for any positive constant  $\varepsilon_1$  there exist a time  $t_1 > 0$  and a bandwidth  $\bar{\omega}_1 \in \mathbb{R}^+$  with the property:

$$\|\tilde{q}_*(t)\| \leq \varepsilon_1 \quad \forall t \geq t_1. \tag{3.20}$$

If Eq. (2.13) is Hurwitz stable, there then exists a positive constant  $k_4$  such that:

$$\left\| y(t) - \left( C \exp(A(t - t_1)) \zeta(t_1) + C \int_{t_1}^t \exp(A(t - \tau)) B u_c(\tau) d\tau \right) \right\| \leq k_4 \varepsilon_1, \tag{3.21}$$

for all  $t \geq t_1$ .

We next prove Lemma 2.

**Proof.** Since  $A_K$  is assumed to be Hurwitz, there exists  $k_0$  and  $\alpha_0$  such that the norm of the time response of the uncertainty error can be expressed as (see Eq. (3.17) and H9):

$$\begin{aligned} \|\tilde{q}_*(t)\| &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + k_2 \|A_K^{-1}\| \int_0^t e^{-\alpha_0(t-\tau)} \left\| \frac{dq_*(\tau)}{d\tau} \right\| d\tau, \\ &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + (k_2/\alpha_0) \mathcal{S}_{\mathcal{R}} \|A_K^{-1}\|, \end{aligned} \tag{3.22}$$

where  $k_1 = k_0 \|G^\ell C\|$  and  $k_2 = k_0 \|G^\ell C\| \|B\|$ . From Theorem II.5.10 of [12] we deduce:

$$\begin{aligned} \|\tilde{q}_*(t)\| &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + (k_2/\alpha_0) \mathcal{S}_{\mathcal{R}} \frac{\|A_K\|^{n-1}}{|\det A_K|}, \\ &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + (k_2/\alpha_0) \mathcal{S}_{\mathcal{R}} \frac{\|A_K\|^{n-1}}{\bar{\omega}_1^n}, \\ &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + (k_3/\alpha_0) \mathcal{S}_{\mathcal{R}} \frac{\varrho_n^{n-1}}{\bar{\omega}_1}. \end{aligned} \tag{3.23}$$

This fact proves (3.19).

Moreover, Eq. (3.20) is a direct consequence of Eq. (3.19) and relation (3.21) follows from Eqs. (3.18) to (3.20), where

$$k_4 = \left\| C \int_{t_1}^{\infty} \exp(A(t - \tau)) B d\tau \right\|. \tag{3.24}$$

□

Lemma 2 makes it possible to consider a disturbance with a band limited spectrum and to establish our next result under the next assumption.

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx

19

**H10.** Assume  $q_*$  is a band-limited function, namely, there exists a  $\mathcal{B}_W > 0$  such that:

$$\mathbf{Q}_*(\omega) = 0, \quad \text{for all } |\omega| > \mathcal{B}_W,$$

with  $\mathbf{Q}_*(\omega)$  as a Fourier transform of  $q_*(t)$ .

**Lemma 3.** Under Assumption **H10**, there exists a positive constant  $k_4$  such that:

$$\|\tilde{q}_*(t)\|_\infty \leq k_1 e^{-(\bar{\omega}_1/2)t} \|\tilde{k}_0\| + \frac{\mathcal{B}_W}{\bar{\omega}_1} k_4 \|q_*(t)\|_\infty, \quad (3.25)$$

where  $\bar{\omega}_1$  is the bandwidth of the Beard–Jones filter (3.2) and  $\mathcal{B}_W$  is the bandwidth of the introduced disturbances signal  $q_*$ .

We next prove Lemma 3.

**Proof.** Using the celebrated Szökefalvi-Nagy’s inequality [32], we immediately deduce that:

$$\|q_*\|_\infty = \sqrt{m} \sqrt[4]{E_* E_{*1}}, \quad (3.26)$$

where  $E_*$  and  $E_{*1}$  are the energies of  $q_*$  and  $dq_*/dt$ . Here:

$$\begin{aligned} E_* &= \int_{-\infty}^{\infty} \|q_*(t)\|^2 dt \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \|\mathbf{Q}_*(\omega)\|^2 d\omega = \frac{1}{2\pi} \int_{-\mathcal{B}_W}^{\mathcal{B}_W} \|\mathbf{Q}_*(\omega)\|^2 d\omega < \infty, \\ E_{*1} &= \int_{-\infty}^{\infty} \left\| \frac{dq_*(t)}{dt} \right\|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega^2 \|\mathbf{Q}_*(\omega)\|^2 d\omega \\ &= \frac{1}{2\pi} \int_{-\mathcal{B}_W}^{\mathcal{B}_W} \omega^2 \|\mathbf{Q}_*(\omega)\|^2 d\omega < \infty. \end{aligned} \quad (3.27)$$

Moreover,  $\|q_*(t)\| \leq \sqrt{m} \sqrt[4]{E_* E_1}$ , for all  $t \in \mathbb{R}$  and the above equality holds if and only if  $q_*(t) = \mathbf{1} \sqrt[4]{E_* E_{*1}} e^{-\sqrt{E_{*1}/E_*}|t|}$ . From Eq. (3.27) we additionally get:

$$E_{*1} \leq \mathcal{B}_W^2 E_*. \quad (3.28)$$

In the same manner we also obtain:

$$\|dq_*/dt\|_\infty = \sqrt{m} \sqrt[4]{E_{*1} E_{*2}}, \quad (3.29)$$

and finally:

$$E_{*2} \leq \mathcal{B}_W^2 E_{*1}, \quad (3.30)$$

where  $E_{*2}$  is the the energy of  $d^2q_*/dt^2$ . One can see that for all  $q_* \neq 0$  we have:

$$\|dq_*/dt\|_\infty \leq \sqrt{m} \sqrt[4]{\mathcal{B}_W^2 E_{*1}^2} = \sqrt{\mathcal{B}_W} \sqrt[4]{E_{*1}/E_*} \|q_*\|_\infty \leq \mathcal{B}_W \|q_*\|_\infty. \quad (3.31)$$

Therefore Eqs. (3.22) and (3.31) imply:

$$\begin{aligned} \|\tilde{q}_*(t)\| &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + k_2 \|A_K^{-1}\| \int_0^t e^{-\alpha_0(t-\tau)} \left\| \frac{dq_*(\tau)}{d\tau} \right\| d\tau, \\ &\leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + (k_2/\alpha_0) \|A_K^{-1}\| \mathcal{B}_W \|q_*\|_\infty. \end{aligned} \quad (3.32)$$

Setting  $\alpha_0 = \bar{\omega}_1/2$ , we finally deduce Eq. (3.25). The proof is completed.  $\square$

Note that it is a usual practice in Electronic Engineering to relate the slew rate  $\|dq_*/dt\|_\infty$  with the corresponding bandwidth  $\mathcal{B}_W$  (as mentioned in Eq. (3.31)).

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

4. Further extensions of the neighborhood stability concept

We now rewrite the SPFA concept in the case of the particular nonlinear systems of the type (2.13) and prove our next main result.

We assume that the nonlinear perturbation signal  $q(x, u)$  satisfies the following two properties (recall Eqs. (2.4), (2.12) and (2.14)):

**P1.**  $M \mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x, u) = \alpha_M(x)$ . Here  $\alpha_M(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a continuous differentiable vector field and the matrix  $\partial\alpha_M(x)/\partial x|_{x=0}$  has no eigenvalue  $\lambda = -1$ . That means:

$$\|\alpha_M(x_1) - \alpha_M(x_2)\| \leq L_M \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{B}_{\rho_1}, \quad L_M > 0, \quad \rho_1 > 0; \tag{4.1}$$

$$\alpha_M(0) = 0; \tag{4.2}$$

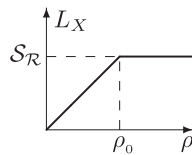
$$\det \left( I + \partial\alpha_M(x)/\partial x|_{x=0} \right) \neq 0. \tag{4.3}$$

**P2.** Let  $X \mathcal{C}_{(M,S)} \Psi_n(d/dt) q(x, u) = \alpha_x(x) + \Gamma_x(x) u$ . Here  $\alpha_x(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and  $\Gamma_x(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$  are continuous differentiable vector fields and  $\alpha_x(x)$  maps the zero vector into the zero vector. The Lipschitz constant of  $\alpha_x(x)$  tends to zero in the neighborhood of the origin and  $\Gamma_x(x)$  has no eigenvalue:  $\lambda = -1$ . Then:

$$\begin{aligned} \|\alpha_x(x_1) - \alpha_x(x_2)\| &\leq L_X \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{B}_\rho, \\ L_X &= \min \left\{ \left( \frac{\mathcal{S}_R}{\rho_0} \right) \rho, \mathcal{S}_R \right\}, \quad \rho \in [0, \rho_1], \end{aligned} \tag{4.4}$$

$$\begin{aligned} 0 < \rho_0 &\leq \rho_1; \\ \alpha_x(0) &= 0; \end{aligned} \tag{4.5}$$

$$\det (I + \Gamma_x(x)) \neq 0 \quad \forall x \in \mathcal{B}_{\rho_1}, \quad \rho_1 > 0. \tag{4.6}$$



**Remark 4.** Under the properties P1 and P2, Eqs. (2.12) and (2.14) take the following respective forms:

$$\zeta = x + \alpha_M(x), \quad q_* = \alpha_x(x) + \Gamma_x(x) u. \tag{4.7}$$

From Eqs. (2.12), (2.13), (2.14) and (4.7), we realize that we are obtaining a similar structure as in Eq. (1.16):  $\zeta = x + \alpha_M(x)$  instead of  $\xi = T_{dif}(x)$  and  $u + \alpha_x(x) + \Gamma_x(x) u$  instead of  $\alpha(x) + \beta(x)u$  (recall also Discussion of Section 1.5.3).

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>



ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx 21

Notice also that in Eq. (2.2), the state space representation  $\Sigma\{A, B, C\}$  is describing the linearized model around the equilibrium point,  $(x, u) = (0, 0)$ , so  $q(x, u) \rightarrow 0$  when  $(x, u) \rightarrow (0, 0)$ , and thus:<sup>10</sup> $q_*(x, u) \rightarrow 0$  when  $(x, u) \rightarrow (0, 0)$ . This phenomenon is characterized by the shape of the Lipschitz constant  $L_X$  (see Eq. (4.4)).

Based on the above results we are now ready to formulate our second main result.

**Theorem 2.** Assume that the nonlinear perturbation signal  $q(x, u)$  satisfies properties P1 and P2. Under the conditions of Lemma 2, we have  $(x, u) = (0, 0) \in \mathcal{X} \oplus \mathcal{U}$  is an equilibrium point of Eq. (2.13). Moreover, there exists a  $\rho_2 > 0$  such that:

$$T_x^\zeta(x) \doteq x + \alpha_M(x), \tag{4.8}$$

is a diffeomorphism that satisfies:

$$\|T_x^\zeta(x)\| \leq (1 + L_M)\|x\|, \quad \text{for all } x \in \mathcal{B}_{\rho_1} \cap \mathcal{B}_{\rho_2}. \tag{4.9}$$

If in addition:

**H11.** For a given  $\mu_0 > 1$  there exists  $\rho_3 > 0$  such that  $\|\alpha_M(x)\| \leq (1 - 1/\mu_0)\|x\|$ , for all  $x \in \bigcap_{i=1}^3 \mathcal{B}_{\rho_i}$ ,

then, the inverse  $T_x^\zeta : T_x^\zeta(\mathcal{B}_{\rho_2}) \rightarrow \mathcal{B}_{\rho_2}$  satisfies the following condition:

$$\|T_x^\zeta(\zeta)\| \leq \mu_0\|\zeta\|, \quad \text{for all } \zeta \in T_x^\zeta(\bigcap_{i=1}^3 \mathcal{B}_{\rho_i}). \tag{4.10}$$

Also, for all non zero  $\zeta$  belonging to  $T_x^\zeta(\mathcal{B}_\rho)$ , where  $\rho \in (0, \rho_0]$ , there holds:

$$\frac{\|q_*(\zeta, 0)\|}{\|\zeta\|} \leq \rho(\mu_0 \mathcal{S}_{\mathcal{R}}/\rho_0). \tag{4.11}$$

Furthermore, if  $A$  is a Hurwitz matrix, then the autonomous system (2.13) is asymptotically stable in the neighborhood  $T_x^\zeta(\mathcal{B}_{\rho_*})$ , where:

$$0 < \rho_* < \min\{\rho_0, \rho_1, \rho_2, \rho_3, \rho_\# \}, \tag{4.12}$$

$$\rho_\# = (2\|B\|\bar{\lambda}\mu_0 \mathcal{S}_{\mathcal{R}}/\rho_0)^{-1}, \tag{4.13}$$

and  $\bar{\lambda}$  is the maximum eigenvalue of  $P$ . Here  $P$  is the positive definite matrix solution of the Lyapunov equation  $A^T P + P A = -I$ .

We next prove Theorem 2, summarized in Fig. 2.

**Proof.** The proof is done in five steps.

1. Let us first prove that  $(x, u) = (0, 0) \in \mathcal{X} \oplus \mathcal{U}$  is an equilibrium point of (Eq. 2.13).

Denote  $p_0 \doteq (0, 0) \in \mathbb{R}^n \times \mathbb{R}^m$  and define the vector field  $f_\zeta(x, u) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  by:

$$f_\zeta(x, u) \doteq A \zeta(t) + B(u(t) + q_*(x(t), u(t))). \tag{4.14}$$

<sup>10</sup> We are also assuming that the time derivative of  $q$  vanish when  $(x, u) \rightarrow (0, 0)$ .

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

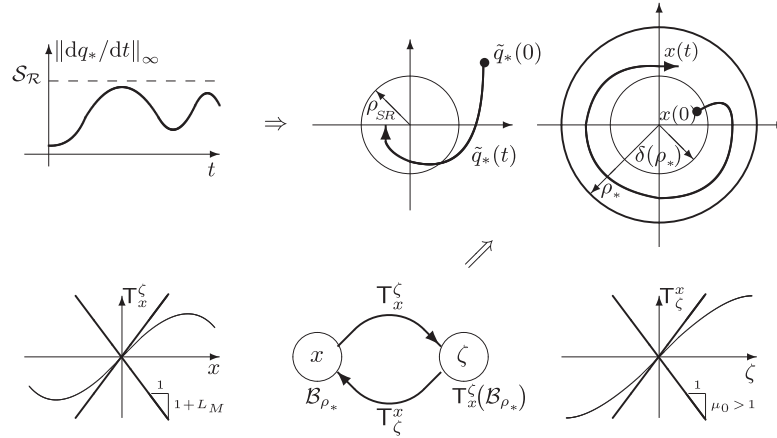


Fig. 2. **Lemma 2** : If  $\|dq_*/dt\|_\infty \leq \mathcal{S}_R$ , then:  $\|\tilde{q}_*(t)\|_\infty \leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + \rho_{SR}$ , where:  $\rho_{SR} = (k_3/\alpha_0) \mathcal{S}_R (e_n^{n-1}/\bar{\omega}_1)$ ;  $\bar{\omega}_1$  is the bandwidth of Eq. (3.2) and  $\mathcal{S}_R$  is the slew rate of  $q_*$ :  $\|dq_*/dt\|_\infty \leq \mathcal{S}_R$ . **Theorem 2**: for all  $x \in \cap_{i=1}^2 \mathcal{B}_{\rho_i}$ ,  $T_x^\zeta(x) \doteq x + \alpha_M(x)$  is a diffeomorphism satisfying:  $\|T_x^\zeta(x)\| \leq (1 + L_M)\|x\|$ . For all  $\zeta \in T_x^\zeta(\cap_{i=1}^3 \mathcal{B}_{\rho_i})$ , the inverse  $T_\zeta^x$  satisfies  $(T_\zeta^x(T_x^\zeta(x)) = x)$ :  $\|T_\zeta^x(\zeta)\| \leq \mu_0 \|\zeta\|$ ,  $\mu_0 > 1$ . If  $A$  is Hurwitz, then Eq. (2.13) is asymptotically stable in  $T_x^\zeta(\mathcal{B}_{\rho_*})$ , where:  $\rho_* < \min\{\rho_0, \rho_1, \rho_2, \rho_3, \rho_\#\}$ ,  $\rho_\# = (2\|B\|\bar{\lambda} \mu_0 \mathcal{S}_R/\rho_0)^{-1}$ .

Substituting Eqs. (4.7) into (4.14), we deduce:

$$f_\zeta(x, u) = A(x + \alpha_M(x)) + B(u + \alpha_x(x) + \Gamma_x(x)u). \tag{4.15}$$

Additionally, from Eqs. (4.2) and (4.5), we also deduce  $f_\zeta(p_0) = 0$ . The last fact implies that  $p_0$  is an equilibrium point of Eq. (2.13).

2. Let us next prove that (Eq. 4.9) is satisfied. Note that from Eqs. (4.8) and (4.3) it follows that the Jacobian matrix  $\partial T_x^\zeta(x)/\partial x$  is nonsingular at  $x = 0$ .

Recall that the classic Inverse Function Theorem implies that there exists a  $\rho_2 > 0$  such that  $T_x^\zeta$  is a diffeomorphism in  $\mathcal{B}_{\rho_2}$  (see e.g., [39], [23],...). Also, from Eqs. (4.8), (4.2) and (4.1), we get:

$$\|T_x^\zeta(x)\| = \|x + \alpha_M(x)\| \leq (1 + L_M)\|x\|.$$

3. Let us now prove that Eq. (4.10) is satisfied. Using Eqs. (4.8), (4.7) and H11, we obtain the series of formal consequences:

$$\begin{aligned} \|T_\zeta^x(\zeta)\| &= \|T_\zeta^x(x + \alpha_M(x))\| = \|T_\zeta^x(T_x^\zeta(x))\| = \|x\| = \|T_x^\zeta(x) - \alpha_M(x)\| \\ &= \|\zeta - \alpha_M(T_x^\zeta(x))\| \leq \|\zeta\| + (1 - 1/\mu_0)\|T_x^\zeta(x)\|. \end{aligned}$$

This fact finally implies Eq. (4.10).

4. Let us prove Eq. (4.11). From Eqs. (4.7), (4.5), (4.4) and (4.10), we get:

$$\frac{\|q_*(\zeta, 0)\|}{\|\zeta\|} = \frac{\|\alpha_x(T_x^\zeta(\zeta))\|}{\|\zeta\|} \leq L_X \frac{\|T_x^\zeta(\zeta)\|}{\|\zeta\|} \leq L_X \mu_0 \leq \bar{\rho}(\mu_0 \mathcal{S}_R/\rho_0).$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx 23

5. Let us finally prove the local asymptotic stability of the autonomous system (2.13). Introduce the Lyapunov function:

$$V(\zeta(t)) = \zeta^T(t)P\zeta(t),$$

with  $P$  the positive definite matrix solution of the associated Lyapunov equation:

$$A^T P + PA = -I.$$

From Eqs. (2.13) and (4.11) we finally obtain for all non zero  $\zeta$  belonging to  $\mathbb{T}_x^\zeta(\mathcal{B}_{\rho_*})$  (recall Eqs. (4.12) and (4.13)):

$$\begin{aligned} dV(\zeta(t))/dt &= \zeta^T(t)(A^T P + PA)\zeta(t) + 2(Bq_*(\zeta(t), 0))^T P\zeta(t), \\ &= -\|\zeta(t)\|^2 \left(1 - \frac{2\|B\|\|q_*(\zeta, 0)\|\|P\|}{\|\zeta(t)\|}\right), \\ &\leq -\|\zeta(t)\|^2(1 - 2\|B\|\rho_*(\mu_0\mathcal{S}_{\mathcal{R}}/\rho_0)\bar{\lambda}) \\ &= -\|\zeta(t)\|^2(1 - \rho_*/\rho_*) < 0. \end{aligned}$$

The proof is now completed.  $\square$

The above result is the SPFA version for the case of particular nonlinear systems of the type (2.13) under the structural assumptions of Theorem 2.

Let us discuss shortly the possible conservativeness of the obtained theoretical result [16,17,23]. Theorem 2 is proven under specific assumptions for the signal  $q(x, u)$ , namely, assumptions P1 and P2. The condition P1 constitutes in fact the smoothness of the signal under consideration, the (global) Lipschitz property and the non-singularity hypothesis associated with the second derivative of the signal. Taking into consideration these natural analytic requirements one can characterize P1 as a non-restrictive condition. The second assumption related to the signal  $q(x, u)$  in Theorem 2 (assumption P2) additionally implies a weak dependence on the state variable  $x$  at the origin. Evidently, this condition in combination with the assumed linear form of the derivative restricts the universality of the obtained result. However, taking into consideration a wide class of signals that satisfy the above condition P1 and P2 one can conclude that the developed theory is in some sense conservative for the class of signals under consideration. Since the proof of Theorem 2 uses explicitly the above mentioned assumptions, the ‘‘sensitivity’’ of the obtained robustness theory is a subject for future research.

We next deduce an immediate consequence for the system (2.13) fed back by Eqs. (3.3) and (3.2), namely for the closed loop system represented by Eq. (3.16).

**Corollary 1.** *Let us consider the autonomous closed loop representation (3.16) satisfying the conditions of Theorem 2 and Lemma 3. Then, the nonlinear uncertainty signals  $q_*(x)$  and  $\tilde{q}_*(x)$  of the open loop system (2.13) and the closed loop system (3.16) are related as follows:*

$$\|q_*(x)\| \leq \gamma_1 \|\alpha_x(x)\| + \gamma_2 \|\tilde{q}_*(x)\|, \quad \forall x \in \mathcal{B}_{\rho_1}, \tag{4.16}$$

where  $\gamma_1 = \sup_{x \in \mathcal{B}_{\rho_1}} \|(I + \Gamma_x(x))^{-1}\|$  and  $\gamma_2 = \sup_{x \in \mathcal{B}_{\rho_1}} \|(I + \Gamma_x(x))^{-1}\Gamma_x(x)\|$ .

Moreover, if the bandwidth  $\bar{\omega}_1$  of the Beard–Jones filter (3.2) additionally satisfies:

$$\bar{\omega}_1 > \mathcal{B}_W k_4 \gamma_2, \tag{4.17}$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

where  $\mathcal{B}_W$  is the bandwidth of  $q_*$ , then:

$$\|\tilde{q}_*(\zeta(t))\|_\infty \leq \frac{k_1 \bar{\omega}_1 / \mathcal{B}_W}{(\bar{\omega}_1 / \mathcal{B}_W - k_4 \gamma_2)} \|\tilde{k}_0\| e^{-\alpha_0 t} + \frac{k_4 \mu_0 \gamma_1}{(\bar{\omega}_1 / \mathcal{B}_W - k_4 \gamma_2)} L_X \|\zeta(t)\|_\infty, \quad (4.18)$$

for all  $\zeta \in \mathbb{T}_x^\zeta(\cap_{i=1}^3 \mathcal{B}_{\rho_i})$ .

Furthermore, given  $\underline{\rho}_{**}$  and  $\bar{\rho}_{**}$  such that  $0 < \underline{\rho}_{**} < \bar{\rho}_{**} < \min\{\rho_1, \rho_2, \rho_3\}$ , the time derivative of the Lyapunov function:

$$V(\zeta(t)) = \zeta^T(t) P \zeta(t),$$

where  $P$  is the positive definite matrix solution of the associated Lyapunov equation:

$$A^T P + P A = -I,$$

satisfies for all  $\zeta(t) \in \mathbb{T}_x^\zeta(\mathcal{B}_{\bar{\rho}_{**}}) \setminus \mathbb{T}_x^\zeta(\mathcal{B}_{\underline{\rho}_{**}})$  and all  $t > t_{\#\#}$ :

$$\frac{d}{dt} V(\zeta(t)) \leq -\left( (1 + L_M) \underline{\rho}_{**} \right)^2 \left( 1 - \frac{\rho_{\#\#} - k_4 \gamma_2}{\bar{\omega}_1 / \mathcal{B}_W - k_4 \gamma_2} \right), \quad (4.19)$$

where  $\bar{\lambda}$  is the maximum eigenvalue of  $P$ ,  $\alpha_0$  is set to  $\bar{\omega}_1/2$ , and:

$$\rho_{\#\#} = 4 \|B\| \bar{\lambda} (k_4 \mu_0 \gamma_1) L_X \left( \frac{\bar{\rho}_{**}}{\underline{\rho}_{**}} \right)^2 + k_4 \gamma_2, \quad (4.20)$$

$$t_{\#\#} = \frac{2}{\bar{\omega}_1} \log \left( \frac{(k_1 \bar{\omega}_1 / \mathcal{B}_W) \|\tilde{k}_0\|}{(k_4 \mu_0 \gamma_1) L_X \bar{\rho}_{**}} \right). \quad (4.21)$$

Finally, if the bandwidth  $\bar{\omega}_1$  of the Beard–Jones filter (3.2) satisfies:

$$\bar{\omega}_1 > \mathcal{B}_W \rho_{\#\#}, \quad (4.22)$$

then any trajectory  $\zeta(t)$  inside  $\mathbb{T}_x^\zeta(\mathcal{B}_{\bar{\rho}_{**}})$  is attracted towards the neighborhood  $\mathbb{T}_x^\zeta(\mathcal{B}_{\underline{\rho}_{**}})$ , for  $t > t_{\#\#}$ .

We next prove Corollary 1, summarized in Fig. 3.

**Proof of Corollary 1.** The proof is done in four steps.

1. Let us first prove that  $q_*(x)$  and  $\tilde{q}_*(x)$  are related by Eq. (4.16). From Eqs. (2.14), P2, (3.3), (3.5(b)) and (3.16(c)) we deduce:

$$q_*(x) = \alpha_x(x) + \Gamma_x(x) (\tilde{q}_*(x) - q_*(x)).$$

Then  $(I + \Gamma_x(x))q_*(x) = \alpha_x(x) + \Gamma_x(x)\tilde{q}_*(x)$ . Relation Eq. (4.6) implies

$$q_*(x) = (I + \Gamma_x(x))^{-1} \alpha_x(x) + (I + \Gamma_x(x))^{-1} \Gamma_x(x) \tilde{q}_*(x), \quad \forall x \in \mathcal{B}_{\rho_1},$$

which finally leads to Eq. (4.16).

2. Let us next prove that  $\tilde{q}_*$  satisfies Eq. (4.18). From relations (3.25) and (4.16) we also get:

$$\|\tilde{q}_*(x(t))\|_\infty \leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + \frac{\mathcal{B}_W}{\bar{\omega}_1} k_4 \left( \gamma_1 \|\alpha_x(x(t))\|_\infty + \gamma_2 \|\tilde{q}_*(x(t))\|_\infty \right),$$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx

25

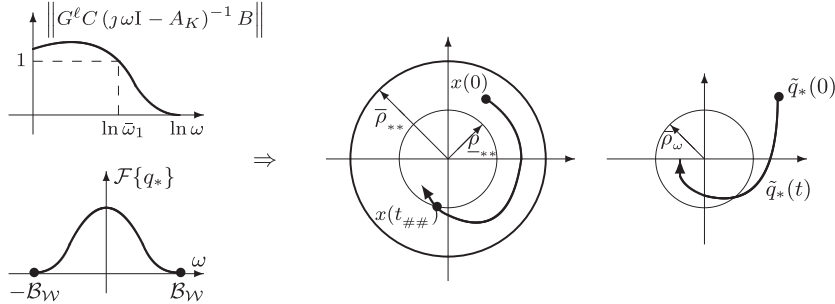


Fig. 3. **Lemma 3** : If  $\mathcal{F}\{q_*\} = 0$  for all  $|\omega| > \mathcal{B}_{\mathcal{W}}$ , then:  $\|\tilde{q}_*(t)\|_{\infty} \leq k_1 e^{-(\bar{\omega}_1/2)t} \|\tilde{k}_0\| + \rho_{\omega}$ , where:  $\rho_{\omega} = (\mathcal{B}_{\mathcal{W}}/\bar{\omega}_1) k_4 \|q_*(t)\|_{\infty}$ ;  $\bar{\omega}_1$  is the bandwidth of Eq. (3.2) and  $\mathcal{B}_{\mathcal{W}}$  is the bandwidth of  $q_*$  (see also (3.24)). **Corollary 1**: If  $\bar{\omega}_1 > \mathcal{B}_{\mathcal{W}} \rho_{\#\#}$ , then any  $x(t) \in \mathcal{B}_{\bar{\rho}_{**}}$  is attracted towards the neighborhood  $\mathcal{B}_{\bar{\rho}_{**}}$  for  $t > t_{\#\#}$ . If  $\bar{\omega}_1 > \mathcal{B}_{\mathcal{W}} k_4 \gamma_2$ , then:  $\|\tilde{q}_*(t)\|_{\infty} \leq \bar{k}_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + \bar{\rho}_{\omega}$ , where:  $\bar{k}_1 = k_1 (\bar{\omega}_1/\mathcal{B}_{\mathcal{W}}) / (\bar{\omega}_1/\mathcal{B}_{\mathcal{W}} - k_4 \gamma_2)$  and  $\bar{\rho}_{\omega} = (k_4 \mu_0 \gamma_1 / (\bar{\omega}_1/\mathcal{B}_{\mathcal{W}} - k_4 \gamma_2)) L_X \|\zeta(t)\|_{\infty}$ .

and:

$$\left(1 - \frac{\mathcal{B}_{\mathcal{W}}}{\bar{\omega}_1} k_4 \gamma_2\right) \|\tilde{q}_*(x(t))\|_{\infty} \leq k_1 e^{-\alpha_0 t} \|\tilde{k}_0\| + \frac{\mathcal{B}_{\mathcal{W}}}{\bar{\omega}_1} k_4 \gamma_1 \|\alpha_x(x(t))\|_{\infty},$$

for all  $x \in \cap_{i=1}^3 \mathcal{B}_{\rho_i}$ . From Eqs. (4.17), (4.4), (4.5) and (4.10), we simply obtain Eq. (4.18).

3. From Eqs. (3.16(a)) and (4.18), we obtain that for all  $\zeta \in \mathbb{T}_x^{\zeta}(\cap_{i=1}^3 \mathcal{B}_{\rho_i})$ :

$$\begin{aligned} \frac{d}{dt} V(\zeta(t)) &= \zeta^T(t) (A^T P + P A) \zeta(t) + 2 (B \tilde{q}_*(\zeta(t)))^T P \zeta(t), \\ &= -\|\zeta(t)\|^2 + 2 (B \tilde{q}_*(\zeta(t)))^T P \zeta(t), \\ &\leq -\|\zeta(t)\|^2 + 2 \|B\| \bar{\lambda} \|\tilde{q}_*(\zeta(t))\| \|\zeta(t)\|, \\ &\leq -\|\zeta(t)\|^2 + 2 \|B\| \bar{\lambda} \left( \frac{k_1 \bar{\omega}_1/\mathcal{B}_{\mathcal{W}}}{(\bar{\omega}_1/\mathcal{B}_{\mathcal{W}} - k_4 \gamma_2)} \right) \|\tilde{k}_0\| e^{-\alpha_0 t} \|\zeta(t)\|_{\infty} \\ &\quad + 2 \|B\| \bar{\lambda} \left( \frac{k_4 \mu_0 \gamma_1}{(\bar{\omega}_1/\mathcal{B}_{\mathcal{W}} - k_4 \gamma_2)} \right) L_X \|\zeta(t)\|_{\infty}^2, \\ &\leq -\|\zeta(t)\|^2 + \frac{2 \|B\| \bar{\lambda}}{\bar{\omega}_1/\mathcal{B}_{\mathcal{W}} - k_4 \gamma_2} \left( \left( k_1 \frac{\bar{\omega}_1}{\mathcal{B}_{\mathcal{W}}} \right) \|\tilde{k}_0\| e^{-\alpha_0 t} \right. \\ &\quad \left. + (k_4 \mu_0 \gamma_1) L_X \|\zeta(t)\|_{\infty} \right) \|\zeta(t)\|_{\infty}. \end{aligned}$$

Setting  $\alpha_0$  to  $\bar{\omega}_1/2$  and considering trajectories of  $\zeta$  inside  $\mathbb{T}_x^{\zeta}(\mathcal{B}_{\bar{\rho}_{**}}) \setminus \mathbb{T}_x^{\zeta}(\mathcal{B}_{\rho_{**}})$ , we get Eq. (4.19) (recall Eq. (4.9)).

4. Let us finally prove that the trajectories of  $\zeta$  inside  $\mathbb{T}_x^{\zeta}(\mathcal{B}_{\bar{\rho}_{**}})$  are attracted towards  $\mathbb{T}_x^{\zeta}(\mathcal{B}_{\rho_{**}})$ . From Eqs. (4.19) and (5.21) we get the negativeness of  $dV(\zeta(t))/dt$ .

This completes the proof.  $\square$

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

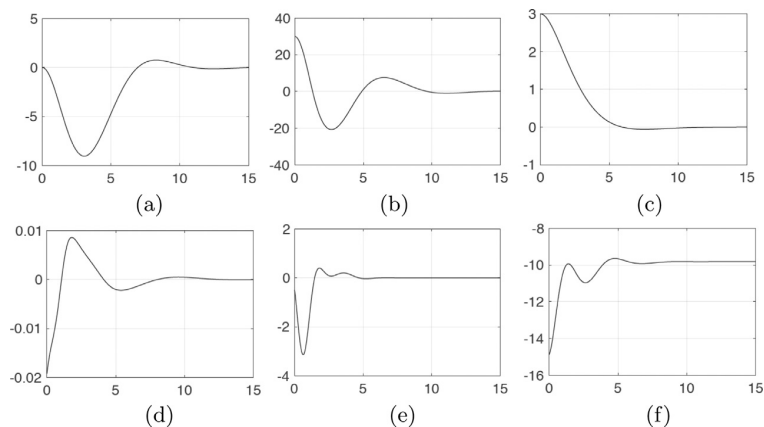


Fig. 4. Nonlinear description (1.4) controlled by the usual exact feedback linearization (1.3) and (1.17). (a)  $x$ [m], (b)  $\theta$ [ $^\circ$ ], (c)  $z$ [m], (d)  $u_x$ [rad kg m s $^{-2}$ ], (e)  $\alpha(x)$ [m s $^{-4}$ ], (f)  $\beta(x)$ [m rad $^{-1}$  s $^{-2}$ ].

For Corollary 1 one can establish similar conservativeness observations as for Theorem 2. We can additionally observe that the (useful) constructive estimation (4.18) is in fact a consequence of the conditions P1 and P2 and assumption (4.17). In that case the sensitive condition (4.17) has a crucial character for the resulting estimation (4.18).

Also, note that:

1. For the case of the *approximated input–output linearization*, we can see from Theorem 2 that the stability of system (2.13) (namely system (2.2)) is only guaranteed for the narrow neighborhood  $\rho_*$  (cf. Eqs. (4.12) and (4.13)). See Fig. 2.
2. For the case of the *robust asymptotic feedback linearization*, we can realize from Corollary 1 that the stability neighborhood is widely enlarged when the *nonlinear uncertainty estimator* (3.3) and (3.2) is added (cf. Eq. (3.16)). Indeed, for a given behavioral neighborhood  $\mathcal{B}_{\bar{\rho}_{**}}$ , a desired attractor neighborhood  $\mathcal{B}_{\rho_{**}}$  is reached by means of a sufficiently large bandwidth  $\bar{\omega}_1$  of the Beard–Jones filter (3.2) (cf. (4.20)). Moreover, with the bandwidth filter  $\bar{\omega}_1$ , we can sufficiently extend the usual linearity neighborhood  $\rho_*$  to  $\rho_{\#\#}$ , and reduce the regular transitory time  $t_{\#\#}$ . See Fig. 3.

Evidently, the presented results not only establish robustness of the resulting system closed by the proposed control law, but also some structural relations between systems under consideration.

## 5. Numerical aspects

We now deal with the computational implementation of the proposed control methodology and consider the main Academic Example 1. Let us note that numerical study presented in this section constitutes a necessary implementability test for the resulting control design algorithm and is not related to any practical system. The numerical results discussed in this section are obtained using the standard MATLAB@packages.

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

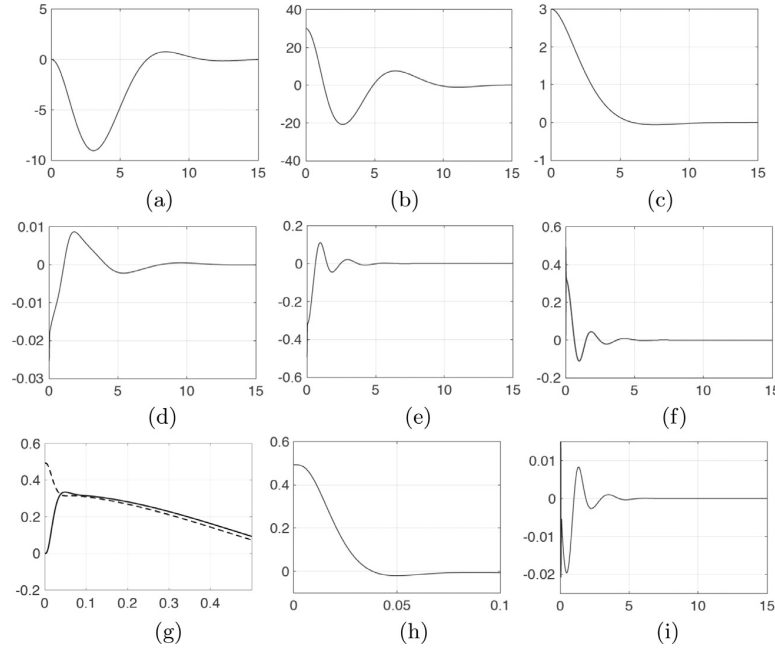


Fig. 5. Nonlinear description (1.2) controlled by the robust asymptotic feedback linearization (1.3), (1.11) and (3.8). (a)  $x$ [m], (b)  $\theta$ [ $^\circ$ ], (c)  $z$ [m], (d)  $u_x$ [ $\text{kg m s}^{-2}$ ], (e)  $q_{x,*}(x)$ [ $\text{s}^{-2}$ ], (f) and (g)  $q_{x,*}(x)$  (dash line) vs  $\bar{u}_x$  (solid line) [ $\text{rad s}^{-2}$ ], (h) and (i)  $q_{x,*}(x) - \bar{u}_x$ [ $\text{rad s}^{-2}$ ].

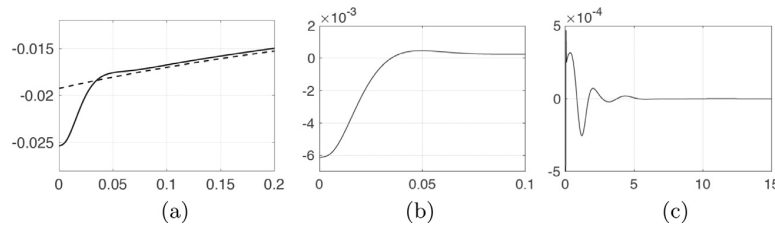


Fig. 6. Comparison of control outputs  $u_x$  obtained in both simulations. (a)  $u_x$  from Fig. 4 (obtained with the exact feedback linearization (1.3) and (1.17); dash line) vs  $u_x$  from Fig. 5 (obtained with the robust asymptotic feedback linearization (1.3), (1.11) and (3.8); solid line) [ $\text{rad kg m s}^{-2}$ ], (b) and (c) Difference of the control signals  $u_x$  issued from Figs. 4 and 5.

Consider the following concrete numerical values of the system parameters:

$$m_T = 0.60 \text{ kg}, \quad J = 0.0032 \text{ kg m}^2, \quad L = 0.17 \text{ m}, \quad g = 9.81 \text{ m s}^{-2}.$$

The earth axes ( $oxyz$ ) is located at height  $\bar{z} = 3.00$  [m] with respect to the ground level. Moreover, we examine the following adequate initial conditions in the framework of the Academic Example 1:  $x(0)=0$ [m],  $z(0)=\bar{z}$ ,  $dx(0)/dt=dz(0)/dt=0$ [ $\text{m s}^{-1}$ ],  $\theta(0)=\theta_0$ ,  $\theta_0=\pi/6$ [rad],  $d\theta(0)/dt=0$ [ $\text{rad s}^{-1}$ ].

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

5.1. State feedbacks

The state feedbacks (1.3) and (1.11) are computed using the algebraic Riccati equations:<sup>11</sup>  $A_{c_2}^T P_2 + P_2 A_{c_2} - (1/5) P_2 B_z B_z^T P_2 + I_2 = 0$  and  $A_{c_4}^T P_4 + P_4 A_{c_4} - (1/5) P_4 B_x B_x^T P_4 + I_4 = 0$ . Here  $F_x = [-a_{x,4} \quad -a_{x,3} \quad -a_{x,2} \quad -a_{x,1}] = -5 B_x^T P_4 = [-0.4472 \quad -1.5814 \quad -2.5725 \quad -2.3119]$ , and  $F_z = [-a_{z,2} \quad -a_{z,1}] = -5 B_z^T P_2 = [-0.4472 \quad -1.0461]$ . The spectra of  $A_x$  and  $A_z$  are as follows  $\Lambda_x(s) = \{-0.4162 \pm 0.7387 i, -0.7398 \pm 0.2736 i\}$  and  $\Lambda_z(s) = \{-0.5231 \pm 0.4167 i\}$ . The spectral radii of  $\Lambda_x(s)$  and  $\Lambda_z(s)$  are  $\rho_x = 0.8479$  and  $\rho_z = 0.6687$ , respectively.

5.2. Beard–Jones filter

Following a root-locus procedure, we get  $s(s + 4.75)(s + 4)(s + 3.5) + 28.125 = (s + 1)^2 (s^2 + 10.25s + 28.125)$ . Implementing the Beard–Jones filter 100 times faster than the spectral radius  $\rho_x$  and  $\rho_{e_x} = 100 \rho_x$ , we obtain  $\bar{K}_x = [-28.125 \rho_{e_x}^4 \quad -66.5 \rho_{e_x}^3 \quad -49.625 \rho_{e_x}^2 \quad -12.25 \rho_{e_x}]^T$ . The initial conditions of Eq. (3.8) were set up as (recall Eqs. (1.23) and (1.20)):  $\bar{w}_x(0) = T_{ox}^{-1} \zeta_x(0)$ , such that:

$$\bar{w}_x(0) = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \\ (1 + a_{z,2} z_0 / g) \tan(\theta_0) \\ -(a_{z,1} a_{z,2} z_0 / g) \tan(\theta_0) \end{bmatrix}.$$

5.3. Simulation results

We first have applied to (1.4) the usual exact feedback linearization procedure (1.3) and (1.17), where  $\alpha(x)$  and  $\beta(x)$  are defined by Eq. (1.21). The state  $\zeta$  is computed based on Eq. (1.20). The coefficients of the polynomials  $\pi_z(s)$  and  $\pi_4(s)$  are determined by the feedbacks  $F_z$  and  $F_x$  (see Fig. 4). We have also applied to Eq. (1.4) the exact structural feedback linearization<sup>12</sup> (1.23) and (1.25). As mentioned above (in Example 2), we have obtained exactly the same time trajectories depicted on Fig. 4. We next apply the structural asymptotic feedback linearization (1.3), (1.11) and (3.8) to the given dynamic model (1.2). The coefficients of the polynomials  $\pi_z(s)$  and  $\pi_{x_x}(s)$  are determined by the feedbacks  $F_z$  and  $F_x$ , respectively (see Fig. 5). Moreover,  $q_{x,*}(x)$  is defined by Eq. (1.25). Notice that the Beard–Jones filter (3.8) quickly tends to the nonlinear uncertainty signal (1.25).

Fig. 6 contains an important comparison between the control outputs  $u_x$  obtained in both simulations. We can note that the robust asymptotic feedback linearization scheme is effective in comparison with the usual exact feedback linearization. On the other hand, the asymptotic scheme does not need any nonlinear signal associated with the derivative computation.

In this real-world oriented example, one cannot analyze correctly the extension of the linearity neighborhood  $\rho_{\#}$ . We refer to an advanced quadrotor model studied in [6] for this consideration. In the presented example we have obtained a good control performance under the perturbation of  $\pi/4$  [rad] with respect to the Euler angles  $(\phi, \theta, \psi)$ . An interesting and important feature of the robust asymptotic feedback linearization scheme that we propose can

<sup>11</sup> We have used the LQR technique, with weighting matrices  $Q = I_2$  and  $R = \rho$  ( $\rho = 5$ ), because one can obtain a good closed-loop behavior by tuning only one parameter,  $\rho$ . In [7], we show some experimental results obtained in open field, and we give more details for a good selection of the weight matrix  $Q$ .

<sup>12</sup> Together with the stabilizing feedback (1.11).

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>



be expressed as follows: it takes into account the non-modeled dynamics; see [5] for the theoretic and experimental results. Also, in [7] is presented a detailed synthesis procedure for an experimental quadrotor laboratory prototype, where some simulations are shown, as well as an experimental proof in open field.

The numerical results presented in this section illustrate the implementability and effectiveness of the proposed structural asymptotic feedback linearization technique. The robust feedback control strategy we developed possesses a computationally easy structure (in comparison to the general PID). Moreover, we does not use here some numerically heavy design techniques, for example, the common LMIs or KLFs approaches. From the other side, the linearization approach we propose has an universal nature and makes it possible to combine the developed robust control technique with the existing (linearization-based) design approaches.

In [7], we have recently presented a paper showing in detail the synthesis procedure of our control scheme proposition in a quadrotor laboratory prototype, in open field. We have tested the effectiveness and simplicity of our proposition with such a quadrotor, in hover flying.

## 6. Conclusion

In this paper, we propose a *robust structural feedback linearization* approach based on failure detection techniques. We consider the non-linear affine system (1.1), and describe it equivalently by the specific state space representation (2.2), together with Eqs. (2.1), (1.7) and (1.6). The newly obtained equivalent system's non-linearities are correspondingly determined by a generic disturbance signal  $q$ . The controllability condition for the pair  $(A, B)$  makes it possible to treat constructively the derived structural algebraic equation (2.4). The last one defines in fact the necessary change of variable (2.12), that also maps the non-linear disturbance signal  $q$  under consideration into a nonlinear uncertainty signal  $q_*$ , contained in the image of  $B$ ; see Eqs. (2.13) and (2.14).

The proposed equivalent “rewriting” technique makes it finally possible to *design a robust feedback* and to *prove the asymptotic stability* of the initially given dynamic model. The abstract mapping mentioned above involves the basic dynamic representation (2.13). This form implies the possibility of the direct “cancellation” (2.16) of  $q_*$ .

If that is not the case, it can be still asymptotically rejected using the Beard–Jones filter (3.2) and (3.3). The main advantage of the developed robust control design is the possibility of the linearity neighborhood extension around the generic equilibrium point 0. Note that the possible uncertainties model is in fact “absorbed” (constructively represented) by  $q_*$ . Additionally, we have obtained a useful theoretical fact, summarized in Corollary 1,

“If the bandwidth  $\bar{\omega}_1$  of the Beard–Jones filter (3.2) satisfies:

$$\bar{\omega}_1 > \mathcal{B}_{\mathcal{W}} \rho_{\#}, \quad (5.21)$$

where  $\mathcal{B}_{\mathcal{W}}$  is the bandwidth of  $q_*$ , then any trajectory  $\zeta(t)$  inside  $\mathbb{T}_x^\zeta(\mathcal{B}_{\bar{\rho}_{**}})$  is attracted towards the neighborhood  $\mathbb{T}_x^\zeta(\mathcal{B}_{\rho_{**}})$ , for  $t > t_{\#}$ ”, namely, the high values of the conventional Beard–Jones filter bandwidth of an uncertain signal extend the necessary linearity neighborhood and sufficiently reduce the transitory time.

Let us discuss shortly some general aspects of the control design methodology we proposed. For example, in comparison with the output feedback recently considered in [43] and [44] the evident advantage of the control approaches expressed by Eqs. (1.11), (2.16) and (3.8) has an “operational nature”. This means that the resulting formulae have a computationally easy structure. The further advantages can be summarized as follows:

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, Journal of the Franklin Institute, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

- the feedback control proposed in our paper is free from the technically sophisticated KLF approach;
- we do not use the “computationally heavy” (in the case of a big system dimension) LMI based techniques;
- the control design approach expressed by the concrete formulae (1.11), (2.16) and (3.8) (as well as some extensions in Section 4) has a simple implementable structure;
- the linearization method we propose constitutes an universal design approach and can be considered in combination with some alternative (linearization-based) control strategies.

Let us finally note that a detailed comparative analysis of the robust control design proposed in our paper and some existing control approaches (for example, with the novel techniques from [43] and [44]) is a subject of a future research.

## References

- [1] S. Ahmed, M.N. Karsiti, G.M. Hassan, Feedback linearized strategies for collaborative nonholonomic robots, in: Proceedings of the International Conference on Control, Automation and Systems, 2007, pp. 1551–1556. Seoul Korea, Oct 17–20
- [2] H. Aling, J.M. Schumacher, A nine-fold canonical decomposition for linear systems, *Int. J. Control* 39 (4) (1984) 779–805.
- [3] J. Back, K.T. Yu, J.H. Seo, Dynamic observer error linearization, *Automatica* 42 (2006) 2195–2200.
- [4] R.V. Beard, Failure Accommodation in Linear Systems Through Self-Reorganization, Massachusetts Institute of Technology, 1971 Ph.D thesis. 1971
- [5] M. Bonilla, L.A. Blas, S. Salazar, J.C. Martínez, M. Malabre, A robust linear control methodology based on fictitious failure rejection, in: Proceedings of the 2016 European Control Conference, 2016, pp. 2596–2601. June 29 – July 1. Aalborg, Denmark.
- [6] L.A. Blas, M. Bonilla, M. Malabre, V. Azhmyakov, S. Salazar, Structural feedback linearization based on nonlinearities rejection, in: Proceedings of the 20th World Congress, 2017, pp. 945–950. July 9–14. Toulouse, France
- [7] L.A. Blas, M. Bonilla, S. Salazar, M. Malabre, V. Azhmyakov, Synthesis of a robust linear structural feedback linearization scheme for an experimental quadrotor, in: Proceedings of the 2019 European Control Conference, 2019, pp. 1431–1436. June 25–28, Napoli, Italy.
- [8] S.A. Bortoff, Approximate state-feedback linearization using spline functions, *Automatica* 33 (8) (1997) 1449–1458.
- [9] P. Brunovsky, A classification of linear controllable systems, *Kybernetika* 6 (3) (1970) 173–188.
- [10] D. Cheng, X. Hu, Y. Wang, Non-regular feedback linearization of nonlinear systems via a normal form algorithm, *Automatica* 40 (2004) 439–447.
- [11] M.V. Cook, Flight Dynamics Principles. A Linear Systems Approach to Aircraft Stability and Control, Elsevier Ltd., New York, 2013.
- [12] C.A. Desoer, M. Vidyasagar, Feedback Systems: Input–Output Properties, Academic Press, New York, 1975.
- [13] D.E. Kim, D.C. Lee, Feedback linearization control of three-phase UPS inverter systems, *IEEE Trans. Ind. Electron.* 57 (3) (2010) 963–968.
- [14] O. Fritsch, P. De Monte, M. Buhll, B. Lohmann, Quasi-static feedback linearization for the translational dynamics of a quadrotor helicopter, in: Proceedings of the 2012 American Control Conference (ACC), 2012, pp. 125–130. Montréal, Canada, June 27–29
- [15] L.R. García Carrillo, A.E.D. López, R. Lozano, C. Pégard, Quad Rotorcraft Control. Vision-Based Hovering and Navigation, Springer-Verlag, London, 2013.
- [16] W.M. Haddad, V. Chellaboina, Dissipativity theory and stability of feedback interconnections for hybrid dynamical systems, *Math. Probl. Eng.* 7 (4) (2001) 299–335.
- [17] W.M. Haddad, V. Chellaboina, S.G. Nersesov, Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity and Control, Princeton University Press, 2014. 2014
- [18] T. Henmi, Y. Park, M. Deng, A. Inoue, Stabilization controller for a cart-type inverted pendulum via a partial linearization method, in: Proceedings of the International Conference on Modelling, Identification and Control, 2010, pp. 248–253. Okayama Japan, July 17–19

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, *Journal of the Franklin Institute*, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

## ARTICLE IN PRESS

JID: FI

[m1+;December 23, 2019;11:47]

*M. Bonilla, L.A. Blas and V. Azhmyakov et al./Journal of the Franklin Institute xxx (xxxx) xxx* 31

- [19] R. Isermann, Process fault detection based on modeling and estimation methods—a survey, *Automatica* 20 (1984) 387–404.
- [20] A. Isidori, A. Ruberti, A relation between different approaches to input–output linearization via feedback, *J. Frankl. Inst.* 320 (6) (1985) 345–350.
- [21] A. Isidori, *Nonlinear Control Systems, An Introduction*, Springer-Verlag, Berlin, 1989.
- [22] T. Kailath, *Linear Systems*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1980.
- [23] H.K. Khalil, *Nonlinear Systems*, Macmillan Publishing Company, New York, 1992. Pearson Education, Prentice Hall (2002)
- [24] K. Komatsu, H. Takata, Nonlinear feedback control of stabilization problem via formal linearization using Taylor expansion, in: *Proceedings of the International Symposium on Information Theory and its Applications*, 2008, 5 pp.
- [25] K.W. Gwak, H.D. Kim, C.W. Kim, Feedback linearization control of a cardiovascular circulatory simulator, *IEEE Trans. Control Syst. Technol.* 23 (5) (2015) 1970–1977.
- [26] A. Mahmood, K. Yoonsoo, Decentralized formation flight control of quadcopters using robust feedback linearization, *J. Frankl. Inst.* 354 (2) (2017) 852–871.
- [27] M.-A. Massoumnia, A geometric approach to the synthesis of failure detection filters, *IEEE Trans. Autom. Control* 31 (9) (1986) 839–846.
- [28] S. Nazrulla, H.K. Khalil, Robust stabilization of non-minimum phase nonlinear systems using extended high-gain observers, *IEEE Trans. Autom. Control* 56 (4) (2011) 802–813.
- [29] H. Nijmeijer, A.J.V.d. Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, New York, 1990.
- [30] G. Oriolo, A. De Luca, M. Vendittelli, WMR control via dynamic feedback linearization: design, implementation, and experimental validation, *IEEE Trans. Control Syst. Technol.* 10 (6) (2002) 835–852.
- [31] G. Palli, C. Melchiorri, T. Wimböck, M. Grebenstein, G. Hirzinger, Feedback linearization and simultaneous stiffness-position control of robots with antagonistic actuated joints, in: *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 4367–4372. Roma, Italy, April 10–14
- [32] A. Papoulis, *Signal Analysis*, McGraw-Hill, New York, 1977.
- [33] J.W. Polderman, J.C. Willems, *Introduction to Mathematical Systems Theory: A Behavioral Approach*, Springer-Verlag, New York, 1998.
- [34] H.H. Rosenbrock, *State-Space and Multivariable Theory*, Nelson, London, 1970.
- [35] J.J.E. Slotine, W. Li, *Applied Nonlinear Control*, Prentice Hall, NY, 1991.
- [36] A. Saberi, A.A. Stoorvogel, P. Sannuti, H.H. Niemann, Fundamental problems in fault detection and identification, *Int. J. Robust Nonlinear Control* 10 (2000) 1209–1236.
- [37] M.W. Spong, M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons, USA, 1989.
- [38] Y. Tang, R.J. Patton, Active FTC for non-linear aircraft based on feedback linearization and robust estimation, in: *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS)*, 2012, pp. 210–215. Mexico, Mexico City, August 29–31
- [39] M. Vidyasagar, *Nonlinear Systems Analysis*, second ed., Prentice-Hall, Inc., New Jersey, 1993.
- [40] J.C. Willems, Input–output and state space representations of finite–dimensional linear time-invariant systems, *Linear Algebra Appl.* 50 (1983) 581–608.
- [41] A.S. Willsky, A survey of design methods for failure detection in dynamic systems, *Automatica* 12 (1976) 601–611.
- [42] W. Zhao, T.H. Go, Quadcopter formation flight control combining MPC and robust feedback linearization, *J. Frankl. Inst.* 351 (3) (2014) 1335–1355.
- [43] Y.W. Member, J. Qiu, H.R. Karimi, Fuzzy-affine-model-based memory filter design of nonlinear systems with time-varying delay, *IEEE Trans. Fuzzy Syst.* 26 (2) (2018) 504–517.
- [44] Y.W. Member, J. Qiu, P. Shi, L. Wu, A piecewise-Markovian Lyapunov approach to reliable output feedback control for fuzzy-affine systems with time-delays and actuator faults, *IEEE Trans. Cybern.* 48 (9) (2018) 2723–2735.
- [45] W.M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 3rd ed., Springer-Verlag, New York, 1985.

Please cite this article as: M. Bonilla, L.A. Blas and V. Azhmyakov et al., Robust structural feedback linearization based on the nonlinearities rejection, *Journal of the Franklin Institute*, <https://doi.org/10.1016/j.jfranklin.2019.11.044>

## Synthesis of a robust linear structural feedback linearization scheme for an experimental quadrotor

L.A. Blas<sup>1</sup>, M. Bonilla<sup>1</sup>, S. Salazar<sup>2</sup>, M. Malabre<sup>3</sup> and V. Azhmyakov<sup>4</sup>

**Abstract**—In this paper, we show in detail a synthesis procedure of the control scheme recently proposed in [3]. This control scheme has the advantage of combining the classical linear control techniques with the sophisticated robust control techniques. This control scheme is specially *ad hoc* for unmanned aircraft vehicles, where it is important not only to reject the actual nonlinearities and the unexpected changes of the structure, but also to look for the simplicity and effectiveness of the control scheme.

### I. INTRODUCTION

Recently, there has been a great interest in finding simple and effective control schemes for unmanned aircraft vehicles, able to reject the actual nonlinearities and the unexpected changes of structure. In [6], the authors present two types of nonlinear controllers for an autonomous quadrotor helicopter. However, to program this control scheme in the embedded autopilot of the quadrotor is rather difficult due to the needed derivative estimations. In [13], the authors present attitude and trajectory tracking control designs based on an inner/outer-loop control structure for normal flight conditions. Once again, to program this control scheme in the embedded autopilot of the quadrotor is rather difficult due to the needed derivative estimations. In [7], the authors propose a robust controller based on the time-scale separation approach to achieve the automatic take-off, hovering, trajectory tracking, and landing missions for a quadrotor helicopter. The authors only show experimental results in indoor environments.

In this paper, we present a synthesis procedure of the recently robust linear control scheme proposed in [3], which is based on failure detection techniques. Such a linear control approach is intended to reject linearly structured uncertainties, which are treated as failure signals affecting the systems dynamics. The implementability and efficiency of the proposed robust control methodology is illustrated with a quadrotor laboratory prototype in hover flying. We present simulation results and experimental results in open field (outdoor environments).

\*The Ph.D. Student Luis Angel Blas Sanchez is sponsored by CONACyT México.

<sup>1</sup>L.A. Blas and M. Bonilla are with CINVESTAV-IPN, Control Automático, UMI 3175 CINVESTAV-CNRS. A.P. 14-740. México 07000. [anghellam@hotmail.com](mailto:anghellam@hotmail.com), [mbonilla@cinvestav.mx](mailto:mbonilla@cinvestav.mx)

<sup>2</sup>S. Salazar is with CINVESTAV-IPN, Sistemas Autónomos de Navegación Aérea Y Submarina, UMI 3175 CINVESTAV-CNRS. A.P. 14-740. México 07000. [sergio.salazar.cruz@gmail.com](mailto:sergio.salazar.cruz@gmail.com)

<sup>3</sup>M. Malabre is with CNRS, LS2N, UMR 6004, Nantes, France. [Michel.Malabre@ls2n.fr](mailto:Michel.Malabre@ls2n.fr)

<sup>4</sup>V. Azhmyakov is with Department of Mathematical Sciences, Universidad EAFIT, Medellín, Republic of Colombia. [vazhmyako@eafit.edu.co](mailto:vazhmyako@eafit.edu.co)

In Section II, we describe the quadrotor laboratory prototype. In Section III, we give the necessary preliminaries to synthesize the control scheme recently proposed in [3]. In Section IV, we synthesize the *exact structural feedback linearization* and the *robust asymptotic feedback linearization* for the dynamics of the longitudinal trajectory  $x$  together with the rotational pitch trajectory  $\theta$  of the quadrotor. In Section V, we do some simulation for tuning our control scheme. In Section VI, we test our proposition in open field. In Section VII, we conclude.

### II. QUADROTOR DESCRIPTION

Let us consider a quadrotor where the total mass is  $M_q$ , the moments of inertia about axis  $ox$ ,  $oy$  and  $oz$  are:<sup>1</sup>  $\mathcal{I}_{xx}$ ,  $\mathcal{I}_{yy}$  and  $\mathcal{I}_{zz}$ , and the distance of each rotor with respect to the centre of gravity of the quadrotor is  $L_m$  (see Fig. 4). The motion is referred to a fixed orthogonal axis set (earth axes)  $(oxyz)$ , where  $oz$  points vertically down along the gravity vector  $[0 \ 0 \ g]^T$ , and the origin  $o$  is located at the desired height  $\bar{z}$ , above the ground level.  $\phi$ ,  $\theta$  and  $\psi$  are the Euler angles, roll, pitch and yaw, measured respectively over the axis  $o_Bx_B$ ,  $o_By_B$  and  $o_Bz_B$ ; where  $(o_Bx_By_Bz_B)$  is the body axis system, with its origin  $o_B$  fixed at the centre of gravity of the quadrotor. We denote:  $\eta = [\phi \ \theta \ \psi]^T$ . The quadrotor is represented by the following state space representations (see [2] for details):

a) *State representation of the  $x - \theta$  dynamics:*

$$\frac{d}{dt}x_x = A_x x_x + B_x u_x + S_x q_{ox}, \quad x = C_x x_x, \quad (2.1)$$

where  $x_x = [x \ dx/dt \ \theta \ d\theta/dt]^T$ ,  $q_{ox} = [q_x \ q_\theta]^T$ , and the matrices  $A_x$ ,  $B_x$ ,  $S_x$  and  $C_x^T$  are, respectively,

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/\mathcal{I}_{yy} \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.2)$$

b) *State representation of the  $y - \phi$  dynamics:*

$$\frac{d}{dt}x_y = A_y x_y + B_y u_y + S_y q_{oy}, \quad y = C_y x_y, \quad (2.3)$$

where  $x_y = [y \ dy/dt \ \phi \ d\phi/dt]^T$ ,  $q_{oy} = [q_y \ q_\phi]^T$ , and the matrices  $A_y$ ,  $B_y$ ,  $S_y$  and  $C_y^T$  are, respectively,

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/\mathcal{I}_{xx} \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.4)$$

<sup>1</sup>Since the quadrotor is mechanically symmetric its cross inertia are zero.

c) State representation of the  $\psi$  dynamics:

$$\frac{d}{dt}x_\psi = A_\psi x_\psi + B_\psi u_\psi + S_\psi q_\psi, \quad \psi = C_\psi x_\psi,$$

where  $x_\psi = [\psi \quad d\psi/dt]^T$ , and the matrices  $A_\psi$ ,  $B_\psi$ ,  $S_\psi$  and  $C_\psi^T$  are, respectively,  $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 & 1 \\ 1/\mathcal{I}_{zz} & 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$  and  $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .

d) State representation of the  $z$  dynamics:

$$\frac{d}{dt}x_z = A_z x_z + B_z \Delta u_z + S_z q_z, \quad z = C_z x_z, \quad (2.5)$$

where  $x_z = [z \quad dz/dt]^T$ , and the matrices  $A_z$ ,  $B_z$ ,  $S_z$  and  $C_z^T$  are, respectively,  $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 & 1 \\ 1/M_q & 0 \end{bmatrix}$ ,  $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$  and  $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .

In the fifth Section of [2], we show how the incremental control actions,  $u_x$ ,  $u_y$ ,  $u_z = \Delta u_z - M_q g$  and  $u_\psi$ , are related with the thrusters of the four rotors,  $f_1$ ,  $f_2$ ,  $f_3$  and  $f_4$ . The nonlinear signals,  $q_x$ ,  $q_y$ ,  $q_z$ ,  $q_\phi$ ,  $q_\theta$  and  $q_\psi$ , are:

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \theta - q_{xx} \\ -\phi - q_{yy} \\ -q_{zz} \end{bmatrix} g + \frac{1}{M_q} \begin{bmatrix} q_{xx} \\ q_{yy} \\ q_{zz} \end{bmatrix} \Delta u_z, \quad (2.6)$$

$$\begin{bmatrix} q_\phi \\ q_\theta \\ q_\psi \end{bmatrix} = (\mathbb{J}^{-1}(\eta) - \mathbb{J}^{-1}(0)) \tau - \mathbb{J}^{-1}(\eta) C(\eta, d\eta/dt) \frac{d\eta}{dt}, \quad (2.7)$$

where  $\mathbb{J}$  and  $C(\eta, d\eta/dt)$  are the inertial and the Coriolis matrices, defined in the Appendix of [2], and:<sup>2</sup>

$$\begin{aligned} q_{xx} &= c_\phi s_\theta c_\phi + s_\phi s_\psi, & q_{yy} &= c_\phi s_\theta s_\psi - s_\phi c_\psi, \\ q_{zz} &= c_\phi c_\theta - 1. \end{aligned} \quad (2.8)$$

### III. PRELIMINAIRES

Let us consider a nonlinear system described by the following nonlinear state space representation:

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t) + Sq(x(t), u(t)), \quad y(t) = Cx(t), \quad (3.1)$$

where  $u \in \mathbb{R}^m$  and  $y \in \mathbb{R}^p$  are the input and the output, respectively,  $x \in \mathbb{R}^n$  is the state and  $q \in \mathbb{R}^\mu$  is a nonlinear perturbation signal, here called an uncertainty vector. We do the following assumption:

*H1*: Structural hypothesis:

- 1)  $\ker B = 0$ .
- 2) The pair  $(A, B)$  is controllable, namely (see for example [12]):  $\mathbb{R}^n = \langle A | \text{Im } B \rangle = \text{Im } B + A \text{Im } B + \dots + A^{n-1} \text{Im } B$ .
- 3) The pair  $(C, A)$  is observable.

We need two results stated in [3].

*Lemma 1* ([3]): Under assumptions **H1.1** and **H1.2**, there exist two linear transformations,  $M : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $X : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , solving the matrix equality:

$$AM + BX = I, \quad (3.2)$$

<sup>2</sup>We use the abbreviated notations:  $(c_\phi, c_\theta, c_\psi)$  for  $(\cos \phi, \cos \theta, \cos \psi)$  and  $(s_\phi, s_\theta, s_\psi)$  for  $(\sin \phi, \sin \theta, \sin \psi)$ .

satisfying:  $X = B^\ell(I - AM)$  and  $M^n = 0$ , where  $B^\ell : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is a left inverse of  $B : \mathbb{R}^m \rightarrow \mathbb{R}^n$ ,  $B^\ell B = I$ , and  $M : \mathbb{R}^n \rightarrow \mathbb{R}^n$  has a nilpotent index not greater than  $n$ .

Equality (3.2) directly follows from **H1.2**:  $\mathbb{R}^n = \langle A | \text{Im } B \rangle \subset \text{Im } B + \text{Im } A \subset \mathbb{R}^n$ ; for the other ones see the proof in [3].

Defining:

$$\mathcal{C}_{(M, S)} = [ S \quad MS \quad \dots \quad M^{(n-1)} S ], \quad (3.3)$$

$$\Psi_n(d/dt) = [ I \quad I d/dt \quad \dots \quad I d^{n-1}/dt^{n-1} ]^T, \quad (3.4)$$

and assuming:

*H2*: The subspace  $M \text{Im } S$  is contained in the unobservable subspace  $\langle \ker C | M \rangle = \ker C \cap M^{-1} \ker C \cap \dots \cap M^{-(n-1)} \ker C$ , namely:

$$CMC_{(M, S)} = 0, \quad (3.5)$$

the following Theorem is obtained:

*Theorem 1* ([3]): Given the linear transformations  $M$  and  $X$  from Lemma 1, consider the following change of variable:

$$\zeta(t) = x(t) + MC_{(M, S)} \Psi_n(d/dt) q(x(t), u(t)). \quad (3.6)$$

Under Assumptions **H1.1**, **H1.2** and **H2**, the state representation (3.1) is externally equivalent to the following<sup>3</sup>:

$$\frac{d}{dt}\zeta(t) = A\zeta(t) + B(u(t) + q_*(x(t), u(t))), \quad y(t) = C\zeta(t), \quad (3.7)$$

where the *nonlinear uncertainty signal*,  $q_*$ , is given by:

$$q_*(x(t), u(t)) = XC_{(M, S)} \Psi_n(d/dt) q(x(t), u(t)). \quad (3.8)$$

Theorem 1 is important because it provides us the change of basis (3.6), which aim is to map the uncertainty vector (nonlinear perturbation signal),  $q$ , to the nonlinear uncertainty signal (3.8),  $q_*$ , contained in  $\text{Im } B$ . Thus,

$$u(t) = -q_*(x(t), u(t)), \quad (3.9)$$

exactly linearizes (3.7).

In the case when it is not possible to reconstruct analytically the *nonlinear uncertainty signal*  $q_*$ , or it is a heavy and tedious task to do it, one can still estimate it. For this, we need to add the following assumption:

*H3*: The state space description  $\Sigma(A, B, C)$  (3.1) has no finite invariant zeros at the origin, namely:

$$\text{Im } B \cap A \ker C = 0. \quad (3.10)$$

Assumptions **H1.3** and **H3** make it possible to apply results of [1] and to design a robust disturbance rejection (based on the Beard-Jones filter, see [8] and the reference there included):

$$\begin{aligned} \frac{d}{dt}w(t) &= A_K w(t) - Ky(t) + Bu(t), \\ \hat{q}_*(t) &= -C^\ell(Cw(t) - y(t)), \quad u(t) = -\hat{q}_*(t). \end{aligned} \quad (3.11)$$

<sup>3</sup>Recall that two representations are called externally equivalent if the corresponding sets of all possible trajectories for the external variables expressed in an *input/output partition*  $(u, y)$  are the same [10], [11].

where  $A_K \doteq (A + KC)$  with  $K : \mathbb{R}^p \rightarrow \mathbb{R}^m$  an output injection such that:  $\sigma\{(A + KC)\} \subset \mathbb{C}^-$ , and  $G^\ell$  is a left inverse of the static gain  $-C(A + KC)^{-1}B$ . The remainder generator is expressed as:

$$\frac{d}{dt}e(t) = A_K e(t) - Bq_*(x, u), \quad \dot{q}_*(t) = -G^\ell C e(t), \quad (3.12)$$

where  $e(t) = w(t) - \zeta(t)$ . In the case when the classic Laplace transform of  $q_*(x, u)$  is well-defined, we have the following transfer function:

$$F_e(s) = G^\ell C (sI - A_K)^{-1} B. \quad (3.13)$$

Under the natural boundedness assumption for  $q_*(x, u)$  with a bandwidth  $\omega_q$ , we have to synthesize a Hurwitz low-pass filter  $F_e(s)$  with a corner frequency  $\omega_c$ , which aim is to reconstruct  $q_*(x, u)$ . Indeed, the *nonlinear uncertainty signal*  $q_*$  is affecting the closed loop behavior throw the high-pass filter  $1 - F_e(s)$ , so the corner frequency  $\omega_c$  of the low-pass filter  $F_e(s)$  should be sufficiently greater than the bandwidth  $\omega_q$ , in order to reject  $q_*$ . This is with the aim to achieve a robust disturbance in a neighborhood around the equilibrium point  $(x, u) = (0, 0)$ , namely:

$$\|q_*(\omega) - \hat{q}_*(\omega)\| \leq \|(I - F_e(j\omega)) X C_{(M, S)} \Psi_n(j\omega)\| \|q(\omega)\|.$$

#### IV. LINEAR ROBUST CONTROL SYNTHESIS

In this Section, we synthesize the *exact structural feedback linearization* (3.8) and (3.9), and the *robust asymptotic feedback linearization* (3.11), for the  $x - \theta$  dynamics (2.1) and (2.2); the other ones are done in a very similar way.

##### A. Locally Stabilizing Feedback

Applying the feedback

$$\begin{aligned} u_x &= F_x x_x + \bar{u}_x, \\ F_x &= \begin{bmatrix} a_{x,4} \mathcal{I}_{yy}/g & a_{x,3} \mathcal{I}_{yy}/g & -a_{x,2} \mathcal{I}_{yy} & -a_{x,1} \mathcal{I}_{yy} \end{bmatrix}, \end{aligned} \quad (4.1)$$

to (2.1), one gets:

$$\frac{d}{dt}x_x = A_{F_x} x_x + B_x \bar{u}_x + S_x q_{ox}, \quad \dot{x} = C_x x_x, \quad (4.2)$$

where:

$$A_{F_x} = A_x + B_x F_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 \\ a_{x,4}/g & a_{x,3}/g & -a_{x,2} & -a_{x,1} \end{bmatrix}, \quad (4.3)$$

with the following Hurwitz characteristic polynomial:

$$\pi_{x_x}(s) = \det(sI - A_{F_x}) = s^4 + a_{x,1} s^3 + a_{x,2} s^2 + a_{x,3} s + a_{x,4}. \quad (4.4)$$

##### B. Exact Structural Feedback Linearization

The matrices  $M_x$  and  $X_x$  solving the algebraic equation (cf. (3.2)):  $A_{F_x} M_x + B_x X_x = I_4$ , and the operators (3.3) and

(3.4) are:

$$\begin{aligned} M_x &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1/g & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\ X_x &= \begin{bmatrix} -a_{x,3} \mathcal{I}_{yy}/g & -a_{x,2} \mathcal{I}_{yy}/g & a_{x,1} \mathcal{I}_{yy} & \mathcal{I}_{yy} \end{bmatrix}, \\ C_{(M_x, S_x)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/g & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1/g & 0 & 0 & 0 \end{bmatrix}, \\ \Psi_4(d/dt) &= \begin{bmatrix} I_2 & I_2 d/dt & I_2 d^2/dt^2 & I_2 d^3/dt^3 \end{bmatrix}^T. \end{aligned} \quad (4.5)$$

Note that:  $M_x^4 = 0$  and  $C_x M_x C_{(M_x, S_x)} = 0$ . The change of variable (3.6) is (cf. (4.5)):

$$\zeta_x = x_x - \left(\frac{1}{g}\right) \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ d/dt & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_\theta \end{bmatrix}. \quad (4.6)$$

The nonlinear uncertainty signal (3.8) is (cf. (4.5)):

$$q_{x,*} = -\frac{\mathcal{I}_{yy}}{g} (a_{x,2} + a_{x,1} d/dt + d^2/dt^2) q_x + \mathcal{I}_{yy} q_\theta. \quad (4.7)$$

The state space representation in the new variable  $\zeta_x$  is (cf. (4.6) and (4.2)):

$$\frac{d}{dt}\zeta_x = A_{F_x} \zeta_x + B_x (\bar{u}_x + q_{x,*}), \quad \dot{x} = C_x \zeta_x. \quad (4.8)$$

and its transfer function is:

$$F_{\zeta_x}(s) = C_x (sI - A_{F_x})^{-1} B_x = -\frac{g}{\mathcal{I}_{yy} \pi_{x_x}(s)}. \quad (4.9)$$

Thus, the *exact structural feedback linearization* is:  $\bar{u}_x = -q_{x,*}(x(t), u(t))$ .

##### C. Robust Asymptotic Feedback Linearization

The observer for rejecting the *nonlinear uncertainty signal*  $q_{x,*}$  is (cf. (3.11)):

$$\begin{aligned} \frac{d}{dt}w_x &= (A_{K_x} + B_x G_x^\ell C_x) w_x - (K_x + B_x G_x^\ell) x, \\ \bar{u}_x &= G_x^\ell C_x w_x - G_x^\ell x, \end{aligned} \quad (4.10)$$

where:

$$A_{K_x} = A_{F_x} + K_x C_x, \quad G_x^\ell = -(C_x A_{K_x}^{-1} B_x)^\ell = -\frac{\mathcal{I}_{yy} a_{x,4}}{g},$$

$$K_x = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1} g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{x,4} - a_{x,4} \\ a_{x,3} - a_{x,3} \\ a_{x,2} - a_{x,2} \\ a_{x,1} - a_{x,1} \end{bmatrix}. \quad (4.11)$$

The characteristic polynomial of the remainder generator (3.12) is:

$$\begin{aligned} \pi_{e_x}(s) &= \det(sI - A_{K_x}) \\ &= s^4 + a_{x,1} s^3 + a_{x,2} s^2 + a_{x,3} s + a_{x,4}. \end{aligned} \quad (4.12)$$

The transfer function of (4.10) is:

$$F_{w_x}(s) = \frac{\mathcal{I}_{yy} a_{x_{o,4}} \pi_{x_x}(s)}{g s \bar{\pi}_{w_x}(s)}, \quad (4.13)$$

$$\bar{\pi}_{w_x}(s) = s^3 + a_{x_{o,1}} s^2 + a_{x_{o,2}} s + a_{x_{o,3}}, \quad (4.14)$$

Let us note that (cf. (4.14) and (4.12)):

$$\pi_{e_x}(s) = s \bar{\pi}_{w_x}(s) + a_{x_{o,4}}. \quad (4.15)$$

So, following a root locus procedure, both polynomials,  $\pi_{e_x}(s)$  and  $\bar{\pi}_{w_x}(s)$ , can be made Hurwitz.

#### D. Closed Loop System

Defining the error signal,  $e_x = w_x - \zeta_x$ , we get the closed loop state space representation (cf. (4.8) and (4.10)):

$$\frac{d}{dt} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix} = A_{CL} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix} + B_{CL} q_{x,*}, \quad x = C_{CL} \begin{bmatrix} e_x \\ \zeta_x \end{bmatrix}, \quad (4.16)$$

$$A_{CL} = \begin{bmatrix} A_{K_x} & 0 \\ B_x G_x^t C_x & A_{F_x} \end{bmatrix}, \quad B_{CL} = \begin{bmatrix} -B_x \\ B_x \end{bmatrix}, \quad (4.17)$$

$$C_{CL} = [0 \mid C_x].$$

The transfer functions are (recall (4.12), (4.9) and (4.15)):

$$F_{e_x}(s) = C_x^t C_x (sI - A_{K_x})^{-1} B_x = \frac{a_{x_{o,4}}}{\pi_{e_x}(s)} \quad (4.18)$$

$$F_{CL}(s) = C_{CL} (sI - A_{CL})^{-1} B_{CL} = [0 \mid C_x].$$

$$\begin{bmatrix} \frac{(sI - A_{K_x})^{-1}}{(sI - A_{F_x})^{-1} B_x G_x^t C_x (sI - A_{K_x})^{-1}} \mid \frac{0}{(sI - A_{F_x})^{-1}} \end{bmatrix} \cdot \begin{bmatrix} -B_x \\ B_x \end{bmatrix}$$

$$F_{CL}(s) = F_{\zeta_x}(s) \left(1 - F_{e_x}(s)\right) = -\frac{g}{\mathcal{I}_{yy} \pi_{x_x}(s)} \left(1 - \frac{a_{x_{o,4}}}{\pi_{e_x}(s)}\right) = -\frac{g}{\mathcal{I}_{yy} \pi_{x_x}(s)} \left(\frac{s \bar{\pi}_{w_x}(s)}{\pi_{e_x}(s)}\right), \quad (4.19)$$

The transfer function (4.19) of the closed loop state space representation, (4.16) and (4.17), has incorporated the high-pass filter:

$$G_{\text{HPF}_x}(s) = 1 - F_{e_x}(s) = 1 - a_{x_{o,4}}/\pi_{e_x}(s) = s \bar{\pi}_{w_x}(s)/\pi_{e_x}(s). \quad (4.20)$$

Hence, if the *nonlinear uncertainty signal*  $q_{x,*}$  has a finite bandwidth, it is then sufficient to synthesize the high-pass filter with a cutoff frequency sufficiently higher than the bandwidth of  $q_{x,*}$ .

#### V. NUMERICAL SIMULATIONS

The numerical values of our laboratory prototype are:

$$M_q = 1.36 \text{ [kg]}, \quad g = 9.81 \text{ [m s}^{-2}\text{]}, \quad \mathcal{I}_{xx} = 0.0134 \text{ [kg m}^2\text{]}, \\ \mathcal{I}_{yy} = 0.0140 \text{ [kg m}^2\text{]}, \quad \mathcal{I}_{zz} = 0.0256 \text{ [kg m}^2\text{]}, \quad (5.1)$$

and:  $L_m = 0.245 \text{ [m]}$ ,  $k_\tau = 4.31 \times 10^{-9} \text{ [N m/rpm}^2\text{]}$ ,  $k_f = 1.98 \times 10^{-7} \text{ [N/rpm}^2\text{]}$  and  $\gamma = k_\tau/k_f = 0.0218 \text{ [m]}$ .

#### A. Locally Stabilizing Feedbacks

The state feedbacks were computed by LQR techniques, namely by solving the algebraic Riccati equation,

$$A^T P + P A - P B (\rho I)^{-1} B^T P + Q = 0,$$

with the following matrices choices:

$$Q_x = Q_y = 900 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2.25 \end{bmatrix}, \quad Q_z = \begin{bmatrix} 1 & 0 \\ 0 & 0.23 \end{bmatrix},$$

$$Q_\psi = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}, \quad (\rho_x, \rho_y, \rho_z, \rho_\psi) = (1, 1, \frac{1}{19600}, \frac{1}{12100}).$$

The optimal state feedbacks are:

$$F_x = [30 \quad 32.4264 \quad -171.9158 \quad -45.0535], \\ F_y = [-30 \quad -32.4260 \quad -171.9116 \quad -45.0512], \\ F_z = [-140 \quad -69.92], \quad F_\psi = [-110 \quad -85.2387]. \quad (5.2)$$

The spectrums of  $A_{F_x}$ ,  $A_{F_y}$ ,  $A_{F_z}$  and  $A_{F_\psi}$  are:

$$\Lambda_x(s) = \{-3214, -1.911, -0.9536 \pm 1.585j\}, \\ \Lambda_y(s) = \{-3358, -1.911, -0.9536 \pm 1.585j\}, \\ \Lambda_z(s) = \{-2.087, -49.32\}, \quad \Lambda_\psi(s) = \{-1.291, -3328\}. \quad (5.3)$$

The spectrum sets  $\Lambda_x$  and  $\Lambda_y$  are lower bounded by:  $\varrho_\ell = \sqrt{0.9536^2 + 1.585^2} = 1.85$ . The characteristic polynomial (4.4) is:

$$\pi_{x_x}(s) = s^4 + 3218 s^3 + 12279 s^2 + 22722 s + 21022. \quad (5.4)$$

#### B. Nonlinear Uncertainty Signals Observation

Since the quadrotor is planned to move on a  $(x, y)$ -plan over a fixed altitude,  $z = \bar{z}$ , we only synthesize *nonlinear uncertainty signals* observers for  $q_{x,*}$  and  $q_{y,*}$ .

1) *Observation's dynamics*: The dynamics of the remainder generator (3.12) and the *nonlinear uncertainty signal* observer (4.10) are given by the polynomials (4.12) and (4.14), respectively; these two polynomials are related by (4.15). Following a root locus procedure, we get:<sup>4</sup>

$$\bar{s}(\bar{s} + 4.75)(\bar{s} + 4)(\bar{s} + 3.5) + 28.125 = \\ (\bar{s} + 1)^2 (\bar{s}^2 + 10.25 \bar{s} + 28.125). \quad (5.5)$$

Scaling the polynomials (5.5) by a factor  $\varrho_c$ , 40 times the lower bounded  $\varrho_\ell$ , we get ( $s = \varrho_c \bar{s}$ ,  $\varrho_c = 40 \varrho_\ell = 74$ ):

$$s \bar{\pi}_{w_x}(s) = s(s + 4.75 \varrho_c)(s + 4 \varrho_c)(s + 3.5 \varrho_c) \\ \pi_{e_x}(s) = (s + \varrho_c)^2 (s^2 + 10.25 \varrho_c s + 28.125 \varrho_c^2). \quad (5.6)$$

In Fig 1, we show the Bode plot of the high-pass filter  $G_{\text{HPF}_x}(j\omega)$ , (4.20) and (5.6). The cutoff frequency  $\omega_c$  of the Bode plot, of the high-pass filter  $G_{\text{HPF}_x}(j\omega)$ , (4.20) and (5.6),

<sup>4</sup>The natural frequency and the damping ratio of the second order factor are respectively:  $\omega_n = 5.3 \text{ rad/s}$  and  $\varrho = 0.966$ .

is (recall (4.12), (4.20), (5.5) and (5.6)):<sup>5</sup>

$$\begin{aligned}\omega_c &= \sqrt{a_{x_{o,2}}/2} \sqrt{1 - \sqrt{1 - 2a_{x_{o,4}}/a_{x_{o,2}}^2}} \\ &= 0.5339 \varrho_c = 39.5 \text{ rad/s.}\end{aligned}\quad (5.7)$$

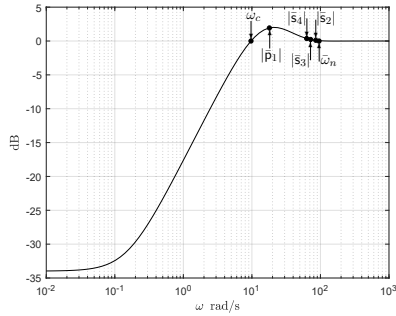


Fig. 1. Bode plot of  $G_{\text{HPF}_x}(j\omega)$ , (4.20) and (5.6).  $|p_1| = 18$ ,  $|s_2| = 85.5$ ,  $|s_3| = 72$ ,  $|s_4| = 63$ ,  $\omega_n = 95.46$  rad/s,  $\zeta = 0.966$  y  $\omega_c = 9.7$  rad/s.

2) *Beard-Jones filter*: For simplicity, the Beard-Jones filter (4.10) is synthesized in its *observer canonical form* [5]:

$$\begin{aligned}\frac{d}{dt} \bar{w}_x &= (\bar{A}_{\bar{K}_x} + \bar{B}_x G_x^\ell \bar{C}_x) \bar{w}_x - (\bar{K}_x + \bar{B}_x G_x^\ell) x, \\ \bar{u}_x &= G_x^\ell \bar{C}_x \bar{w}_x - G_x^\ell x,\end{aligned}\quad (5.8)$$

where:  $\bar{A}_{\bar{K}_x} = \bar{A}_x + \bar{K}_x \bar{C}_x$ , and (recall (5.1), (4.4), (5.4), (4.12) and (5.6)):<sup>6</sup>

$$\bar{A}_x = \begin{bmatrix} 0 & 0 & 0 & -a_{x,4} \\ 1 & 0 & 0 & -a_{x,3} \\ 0 & 1 & 0 & -a_{x,2} \\ 0 & 0 & 1 & -a_{x,1} \end{bmatrix}, \quad (\bar{B}_x, \bar{C}_x, \bar{K}_x) =$$

$$\left( \begin{bmatrix} -g/I_{yy} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T, \begin{bmatrix} a_{x,4} - a_{x_{o,4}} \\ a_{x,3} - a_{x_{o,3}} \\ a_{x,2} - a_{x_{o,2}} \\ a_{x,1} - a_{x_{o,1}} \end{bmatrix} \right),$$

$$G_x^\ell = -(\bar{C}_x \bar{A}_{\bar{K}_x}^{-1} \bar{B}_x)^\ell = -(a_{x_{o,4}} \mathcal{I}_{yy})/g,$$

$$\bar{K}_x = [-843351429 \quad -26924674 \quad -259467 \quad 2311.6]^T. \quad (5.9)$$

### C. Simulation Results

In Fig. 2, we show some simulation results obtained in a MATLAB<sup>®</sup> platform. We have considered that the earth axes

<sup>5</sup>Doing:  $|G_{\text{HPF}_x}(j\omega)| = 1$ , one gets:  $2a_{x_{o,4}}\omega^4 - 2a_{x_{o,2}}a_{x_{o,4}}\omega^2 + a_{x_{o,4}}^2 = 0$ .

<sup>6</sup> $\bar{w}_x = T_{ox}^{-1} w_x$ ,  $T_{ox} = \begin{bmatrix} a_{x,3} & a_{x,2} & -a_{x,1}g & -g \\ a_{x,2} & a_{x,1} & -g & 0 \\ a_{x,1} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1}$ ,  $(\bar{A}_x, \bar{B}_x, \bar{C}_x, \bar{K}_x) = (T_{ox}^{-1} A_x T_{ox}, T_{ox}^{-1} B_x, C_x T_{ox}, T_{ox}^{-1} K_x)$ ,  $G_x^\ell = -(C_x (A_x T_{ox}^{-1} B_x)^{-1} B_x)^\ell = -(\bar{C}_x \bar{A}_{\bar{K}_x}^{-1} \bar{B}_x)^\ell$ .

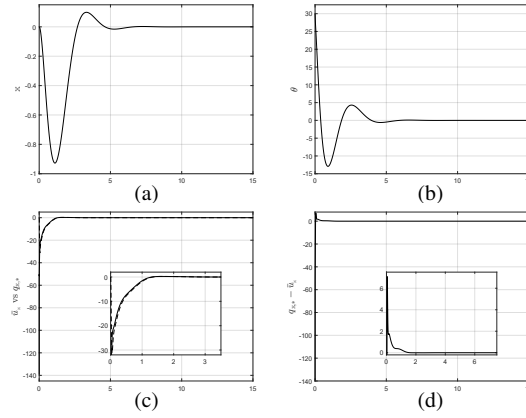


Fig. 2. (a) Horizontal position,  $x$  [m]. (b) Pitch position,  $\theta$  [°]. (c) Estimated nonlinear uncertainty signal (cf. (5.8) and (5.9)),  $\bar{u}_x$  vs computed nonlinear uncertainty signal (cf. (4.7), (2.6)–(2.8)),  $q_{x,*}$ . (d) Estimation error,  $q_{x,*} - \bar{u}_x$ .

( $oxyz$ ) is located at height  $\bar{z} = 2$  [m], above the ground level, and with the initial conditions:  $x(0) = y(0) = z(0) = 0$  [m],  $dx(0)/dt = dy(0)/dt = dz(0)/dt = 0$  [m s<sup>-1</sup>],  $\phi(0) = \theta(0) = \psi(0) = \alpha_0$ ,  $\alpha_0 = \pi/6$  [rad],  $d\phi(0)/dt = d\theta(0)/dt = d\psi(0)/dt = 0$  [rad s<sup>-1</sup>].

The initial conditions of the Beard-Jones filter (5.8) and (5.9) were set up as (cf. [2]):

$$\begin{aligned}\bar{w}_x(0) &= -[a_{x,1}g \quad g \quad 0 \quad 0]^T \tan \alpha_0 (\cos \alpha_0 + \tan \alpha_0), \\ \bar{w}_y(0) &= [a_{y,1}g \quad g \quad 0 \quad 0]^T \tan \alpha_0 (1 - \sin \alpha_0), \\ \bar{w}_\psi(0) &= [\alpha_0 a_{\psi,1} \quad \alpha_0]^T.\end{aligned}\quad (5.10)$$

From Fig. 2, we can verify that the *nonlinear uncertainty signal* estimation  $\bar{u}_x$ , obtained via the Beard-Jones filter (5.8) and (5.9), closely follows the *nonlinear uncertainty signal*  $q_{x,*}$ , computed via (4.7), (2.6)–(2.8).

## VI. EXPERIMENTAL RESULTS

In Fig. 3, we show some experimental results obtained in open field; we compare the horizontal position trajectories  $x$ : (i) when both control laws are applied, the locally stabilizing feedback  $F_x x_x$  (4.1) and (5.2) plus the estimation of the *nonlinear uncertainty signal*  $\bar{u}_x$  (5.8) and (5.9) (solid line), and (ii) when only the locally stabilizing feedback  $F_x x_x$  (4.1) and (5.2) is applied. From this figure, we can point out the correct rejection of the *nonlinear uncertainty signal*  $q_{x,*}$ , via the Beard-Jones filter (5.8) and (5.9).

## VII. CONCLUSION

In [1], we have shown that for a system modeled by the state space representation (3.1), where  $q$  is an *uncertainty signal*, and the map  $S$  is contained in the image of  $B$ , namely there exists a  $Q$  such that:  $S = BQ$ , then the *uncertainty signal*  $q$  can be rejected by means of the Beard-Jones filter (3.11). Such proposition was tested via the altitude control



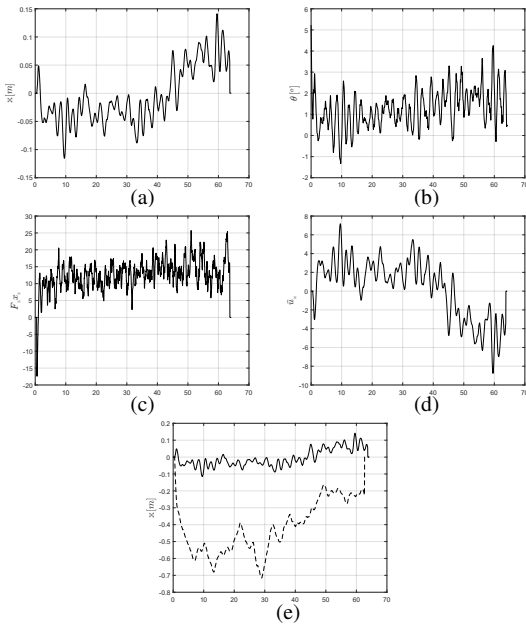


Fig. 3. (a) Horizontal position,  $x$  [m]. (b) Pitch position,  $\theta$  [°]. (c) Locally stabilizing feedback  $F_z x_z$  [N m]. (d) Estimated nonlinear uncertainty signal (cf. (5.8)),  $\bar{u}_x$ . (e) Comparison of the horizontal positions,  $x$  [m], obtained with the application of both, the locally stabilizing feedback  $F_x x_x$  (4.1) and (5.2) plus the estimation of the nonlinear uncertainty signal  $\bar{u}_x$  (5.8) and (5.9) (solid line), against only the application of the locally stabilizing feedback  $F_x x_x$  (4.1) and (5.2) (dashed line).

of a planar vertical takeoff and landing (PVTOL) aircraft in a laboratory setting.

In [2], we have shown that the linear control scheme, based on failure detection techniques, introduced in [1] is in fact a structural feedback linearization technique, where the nonlinearities, affecting the systems dynamics, are treated as failure signals. Based on the Brunovsky canonical form [4], we have proposed the change of variable (3.6), for changing from the state space representation in its Brunovsky canonical form to a nice state space representation, where the uncertainty signal  $q_*$  is inside of the image of the input map, and thus, with a Beard-Jones filter type,  $q_*$  is rejected. This control scheme was illustrated with a MATLAB<sup>®</sup> numerical simulation of a quadrotor in hover flying.

In [3], the two previous results were formalized to the more general nonlinear state space representation (3.1), and we generalized the change of variable (3.6), based on the algebraic equation (3.2), issued from the assumption that the pair  $(A, B)$  is controllable. Under such a change of variable, we have obtained once again the state representation (3.7), where the nonlinear uncertainty signal (3.8) is contained in the image of  $B$ . We have also generalized the previous Beard-Jones filters to the control scheme (3.11).

In this paper, we have shown in detail the synthesis procedure of the control scheme recently proposed in [3]. This

control scheme has the advantage of combining the classical linear control techniques with the sophisticated robust control techniques. This control scheme is specially *ad hoc* for unmanned aircraft vehicles, where it is important not only to have the capability of rejecting the actual nonlinearities and the unexpected changes of structure, but also to obtain simplicity and effectiveness of the control scheme.

We have tested the effectiveness and simplicity of our proposition with the quadrotor laboratory prototype, in hover flying, having the numerical values (5.1). In order to tune our control scheme, we have first done some simulation proofs, shown in Section V, and then we have tested our proposition in open field, as shown in Section VI. Because of lack of place, comparison with alternative approaches will be done in an extended version of the present contribution.

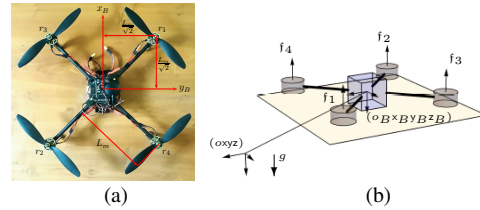


Fig. 4. (a) Quadrotor. (b) Schematic diagram.

REFERENCES

- [1] Bonilla, M., Blas, L.A., Salazar, S., Martínez, J.C. and Malabre, M. (2016). A Robust Linear Control Methodology based on Fictitious Failure Rejection. *European Control Conference*, pp. 2596–2601. June 29 - July 1, 2016. Aalborg, Denmark.
- [2] Blas, L.A., Bonilla, M., Malabre, M., Azhmyakov, V. and Salazar, S. (2017). *IFAC-World Congress*, pp. 2596–2601. July 9 - 14, 2017. Toulouse, France.
- [3] Bonilla, M., Blas, L.A., Azhmyakov, V., Malabre, M. and Salazar, S. (2018). Robust Structural Feedback Linearization Based on the Non-linearities Rejection. *Submitted*, 24 pp.
- [4] Brunovsky, P. (1970). A classification of linear controllable systems. *Kybernetika* 6(3), 173–188.
- [5] Kailath, Th. (1980). **Linear systems**, Prentice-Hall, New Jersey.
- [6] Lee, D., Kim, H.J., Sastry, Sh. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of Control, Automation and Systems* 7(3), pp. 419–428, 2009.
- [7] Liu, H., Bai, Y., Lu, G., Shi, Z., Zhong, Y. Robust Tracking Control of a Quadrotor Helicopter. *Journal of Intelligent & Robotic Systems* 75, pp. 595–608, 2014.
- [8] Massoumnia, M.A. A geometric approach to the synthesis of failure detection filters, *IEEE Transactions on Automatic Control*, 31(9), pp. 839–846, 1986.
- [9] Powers, C., Mellinger, D., and Kumar, V. (2014). Quadrotor Kinematics and Dynamics. In Chapter 16 of **Handbook of unmanned aerial vehicles**, editors Valavanis, K. P., & Vachtsevanos, G. J. New York : Springer Publishing Company, Incorporated.
- [10] Willems, J.C. Input-output and state-space representations of finite-dimensional linear time-invariant systems, *Linear Algebra and its Applications*, 50, 581–608, 1983.
- [11] Willems, J.C. and Polderman, J.W. **Introduction to Mathematical Systems Theory: A Behavioral Approach**, New York: Springer-Verlag, 1998.
- [12] Wonham, W.M. **Linear Multivariable Control: A Geometric Approach**. New York: Springer-Verlag, 3rd ed. 1985.
- [13] Zhou, Q.L., Zhang, Y., Rabbath, C.A., and Theilliol, D. Design of Feedback Linearization Control and Reconfigurable Control Allocation with Application to a Quadrotor UAV. *Conference on Control and Fault Tolerant Systems*, pp. 371–376. October 6 - 8, 2010. Nice, France.

## Structural feedback linearization based on nonlinearities rejection

L.A. Blas<sup>\*,\*</sup> M. Bonilla<sup>\*\*</sup> M. Malabre<sup>\*\*\*</sup> V. Azhmyakov<sup>\*\*\*\*</sup>  
 S. Salazar<sup>†</sup>

\* CINVESTAV-IPN, DCA, México (anghel.lam@hotmail.com).

\*\* CINVESTAV-IPN, DCA, UMI 3175 CINVESTAV-CNRS, México  
 (mbonilla@cinvestav.mx).

\*\*\* CNRS, LS2N (Laboratoire des Sciences du Numérique de Nantes),  
 UMR 6004, France (Michel.Malabre@ls2n.fr).

\*\*\*\* Universidad de Medellín, Department of Basic Sciences, Medellín,  
 Colombia (vazhmyakov@udem.edu.co).

† CINVESTAV-IPN, SANAS, UMI 3175 CINVESTAV-CNRS. México  
 (sergio.salazar.cruz@gmail.com).

**Abstract:** In this paper, a structural feedback linearization technique is proposed. This is a quite simple and effective linear control scheme based on failure detection techniques. Our proposed linear control approach is intended to reject the nonlinearities, which are treated as failure signals affecting the systems dynamics. The proposed control methodology is illustrated via the attitude control of a quadrotor in hover flying.

**Keywords:** Nonlinear systems, feedback linearization, failure reconstruction.

### NOTATION

$\underline{x}_k^i$  stands for the vector in  $\mathbb{R}^k$ , which the  $i$ -th entry is equal to 1 and the others are equal to 0.  $\mathbf{0}_k$  stands for the zero vector in  $\mathbb{R}^k$ .  $\mathbf{I}_k$  stands for the identity matrix of size  $k \times k$ , or shortly  $\mathbf{I}$  when  $k$  does not need to be explicated.  $\mathbf{0}_k$  stands for the zero matrix of size  $k \times k$ .  $\mathbf{0}_{k,\ell}$  stands for the zero matrix of size  $k \times \ell$ .  $\mathbf{N}_k$  stands for the

nilpotent matrix of size  $k \times k$ : 
$$\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix} \cdot \mathbf{C}_k \left( [a_1 \dots a_k] \right)$$

=  $\mathbf{N}_k + \underline{\chi}_k^k [a_1 \dots a_k]$  stands for the companion matrix of size  $k \times k$ .  $\mathbf{BD}\{X_1, \dots, X_k\}$  stands for the block diagonal matrix, where  $X_1, \dots, X_k$  are on the diagonal and the other elements are zero.

### 1. INTRODUCTION

The basic idea of simplifying the form of a nonlinear system by choosing a different state representation has been known in the control community since the work of Isidori (1989) and Nijmeijer and Van der Schaft (1990). However, such a technique is similar to the choice of reference frames or coordinate systems in mechanics. Feedback linearization is equivalent to transforming original system models into equivalent models having a simpler form. In engineering, there are many applications for feedback linearization such as in helicopters, high-performance aircrafts, industrial robots manipulators, biomedical devices, vehicle control.

The central idea of feedback linearization is to algebraically transform nonlinear systems dynamics into (fully or partly) linear ones (see for example Slotine and Li

(1991), Isidori (1989) and Khalil (1992)), so that linear control techniques can be applied.

Let us consider the nonlinear system:

$$dx/dt = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}; \quad \mathbf{y} = \mathbf{C}_o\mathbf{x}, \quad (1.1)$$

where:  $\mathbf{x} \in \mathbb{R}^n$  is the state variable,  $\mathbf{y} = [y_1 \dots y_p]^T \in \mathbb{R}^p$  is the output variable,  $\mathbf{u} = [u_1 \dots u_m]^T \in \mathbb{R}^m$  is the input variable,  $\mathbf{f}(\mathbf{x})$  and  $\mathbf{g}(\mathbf{x})$  are analytic vector fields on  $\mathbb{R}^n$ , and  $\mathbf{C}_o \in \mathbb{R}^{p \times n}$  is an epic matrix, namely:  $\ker \mathbf{C}_o^T = \{0\}$ . We assume that  $p = m$ , and that the system has relative degree  $n$  at any point  $\mathbf{x}_p \in \mathbb{R}^n$ , the set of points where a relative degree can be defined is an open and dense set of the set  $U$  where the System (1.1) is defined (Isidori, 1989; Khalil, 1992; Slotine and Li, 1991).

Let us assume that:

*Hypothesis 1.* The origin  $\mathbf{x}_o = \mathbf{0} \in \mathbb{R}^n$  is an equilibrium point of the autonomous system (1.1), namely:  $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ .

*Hypothesis 2.*  $\mathbf{f}$  and  $\mathbf{g}$  are continuously differentiable.

*Hypothesis 3.* Given the defined matrices,

$$\mathbf{A}_o \doteq [\partial \mathbf{f} / \partial \mathbf{x}]_{\mathbf{x}=\mathbf{0}}, \quad \mathbf{B}_o \doteq [\mathbf{g}]_{\mathbf{x}=\mathbf{0}}, \quad (1.2)$$

(i)  $\ker \mathbf{B}_o = \{0\}$ , (ii) the pair  $(\mathbf{A}_o, \mathbf{B}_o)$  is controllable, and (iii) the pair  $(\mathbf{C}_o, \mathbf{A}_o)$  is observable.

Under these assumptions, it is well known that there exists a stabilizing feedback,  $\mathbf{u}(t) = \mathbf{F}\mathbf{x}(t)$ ,  $\mathbf{F} \in \mathbb{R}^{m \times n}$ , such that:  $\mathbf{A}_o + \mathbf{B}_o\mathbf{F}$  is a Hurwitz matrix, and  $\mathbf{0}$  is an exponentially stable equilibrium point of the autonomous system  $dx/dt = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{F}\mathbf{x}$  (see for example Vidyasagar (1993)). Thus, a common engineering practice is to control (1.1) with standard stabilizing linear control laws. Note

\* The Ph.D. Student Luis Angel Blas Sanchez is sponsored by CONACyT México.

that the stability neighborhood can be small for a control based on a Taylor approximation.

Let us briefly recall the input-output linearization technique for the case:  $p = m = 1$ .

*Hypothesis 4.* The relative degree and the dimension of the state  $\mathbf{x}$  are equal to  $n$ .

Since the system has relative degree  $n$ , exactly equal to the dimension of state space, at some point  $\mathbf{x} = \mathbf{x}_p$ , there then exists a diffeomorphism  $T_{dif}$ ,  $\zeta(t) = T_{dif}(\mathbf{x}(t))$ , such that  $(\zeta = [\zeta_1 \cdots \zeta_n]^T)$ :

$$d\zeta/dt = \mathbf{C}_n (\mathbf{0}_n^T \zeta + \underline{\chi}_n^n(\alpha(x) + \beta(x)u); y(t) = (\underline{\chi}_n^1)^T \zeta$$

Let us note that Hypothesis 4 implies that there exists a known positive constant  $\bar{\beta}$  such that for all  $\mathbf{x}$ :  $0 < \bar{\beta} < \beta(\mathbf{x})$ . Thus, if we could have a complete knowledge of its parameters, then the ideal control law would be:

$$u(t) = \beta^{-1}(\mathbf{x}(t))(-\alpha(\mathbf{x}(t)) + [-a_1 \cdots -a_n] \zeta(t)).$$

With this ideal control law, the closed loop system is described by:

$$d\zeta/dt = \mathbf{C}_n (\underline{\mathbf{a}}^T \zeta(t); y(t) = (\underline{\chi}_n^1)^T \zeta(t),$$

where:  $\underline{\mathbf{a}} = [-a_1 \ -a_2 \ -a_3 \ \cdots \ -a_{n-1} \ -a_n]^T$ .

The feedback linearization problem is achieved by exact state transformation and feedback, rather than by linear approximations of the dynamics. This approach requires a complete knowledge of the model, with exact derivatives, and this is not always possible. There are still a number of shortcomings and limitations associated with the feedback linearization approach. In this paper, we propose a linearization methodology based on the internal structure of a linear part of the system, characterized by the pair  $(A_o, B_o)$ . For this, in Section 2, we first decompose the linear part of (1.1) in its Brunovsky canonical form  $(A_B, B_B)$  (Brunovsky, 1970). And then, in Section 3, we propose an operator  $X(d/dt)$  depending on the structure of the Brunovsky matrix  $A_B$ , which aim is to concentrate the non-linear part of the system in the image of the Brunovsky input matrix  $B_B$ . In Section 4, we asymptotically reject the non-linear components by means of a failure reconstructor. In Section 5, we apply our methodology to a quadrotor in hover flying. In Section 6, we show simulation results. In Section 7, we conclude.

## 2. STRUCTURAL DECOMPOSITION

Let us assume that  $p = m$  and that Hypothesis 1–3 are satisfied. Taking into account (1.2) in (1.1), we get:

$$d\mathbf{x}/dt = A_o \mathbf{x} + B_o \mathbf{u} + \mathbf{f}_o(\mathbf{x}, \mathbf{u}); \mathbf{y} = C_o \mathbf{x}, \quad (2.3)$$

where:  $\mathbf{f}_o(\mathbf{x}, \mathbf{u}) = \Delta \mathbf{f}(\mathbf{x}) + \Delta \mathbf{g}(\mathbf{x})\mathbf{u}$ ,  $\Delta \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - A_o \mathbf{x}$ , and  $\Delta \mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - B_o$ . Let us note that for a given bound,  $\varepsilon_0 \in \mathbb{R}^+$ , there exist a small neighborhood,  $\delta_0 \in \mathbb{R}^+$ , around the equilibrium point,  $\mathbf{0}$ , such that:

$$\|\Delta \mathbf{f}(\mathbf{x})\|_2 \leq \varepsilon_0 \|\mathbf{x}\|_2^2, \|\Delta \mathbf{g}(\mathbf{x})\|_2 \leq \varepsilon_0 \|\mathbf{x}\|_2, \forall \|\mathbf{x}\|_2 < \delta_0$$

Since the pair  $(A_o, B_o)$  is controllable there exist a state feedback  $F_B \in \mathbb{R}^{n \times m}$ , and invertible matrices  $T_B \in \mathbb{R}^{n \times n}$  and  $G_B \in \mathbb{R}^{m \times m}$ , such that (Brunovsky, 1970):

$$d\xi/dt = A_B \xi + B_B \bar{\mathbf{u}} + S_o \mathbf{q}_o(\xi, \bar{\mathbf{u}}); \mathbf{y} = C_B \xi, \quad (2.4)$$

where:

$$\begin{aligned} \mathbf{x} &= T_B \xi, \quad \mathbf{u} = F_B \mathbf{x} + G_B \bar{\mathbf{u}}, \\ A_B &= T_B^{-1} (A_o + B_o F_B) T_B = \mathbf{BD} \{A_1, \dots, A_m\}, \\ B_B &= T_B^{-1} B_o G_B = \mathbf{BD} \{B_1, \dots, B_m\}, \\ C_B &= C_o T_B \doteq [C_1 \cdots C_m], \end{aligned} \quad (2.5)$$

$$\begin{aligned} S_o \mathbf{q}_o &= T_B^{-1} \mathbf{f}_o(\mathbf{x}(t), \mathbf{u}(t)), \quad S_o \doteq [S_1^T \cdots S_m^T]^T, \\ S_i &= [S_{i,1} \cdots S_{i,\ell}], \quad i \in \{1, \dots, m\}, \\ A_i &= \mathbf{C}_{n_i} (\mathbf{0}_{n_i}^T), \quad B_i = \underline{\chi}_{n_i}^m. \end{aligned} \quad (2.6)$$

Thus, we have the following  $m$  pseudo linear state equations perturbed by the nonlinear perturbation signal  $\mathbf{q}_o$ :

$$d\xi_i/dt = A_i \xi_i + B_i \bar{u}_i + S_i \mathbf{q}_o(\xi, \bar{\mathbf{u}}), \quad i \in \{1, \dots, m\}, \quad (2.7)$$

where:  $\bar{\mathbf{u}} \doteq [\bar{u}_1 \cdots \bar{u}_m]^T \in \mathbb{R}^m$ ,  $\xi \doteq [\xi_1^T \cdots \xi_m^T]^T \in \mathbb{R}^n$ ,  $\xi_i \in \mathbb{R}^{n_i}$  ( $i \in \{1, \dots, m\}$ ),  $\sum_{i=1}^m n_i = n$ ,  $\mathbf{q}_o \in \mathbb{R}^\ell$ .

## 3. STRUCTURAL EXACT FEEDBACK LINEARIZATION

Let us define the operator  $X(d/dt)$ , as follows:

$$X(d/dt) = -A_B^T C_n(S_o) \Psi_n(d/dt), \quad (3.8)$$

$$C_\kappa(S_o) = [S_o \ A_B^T S_o \ \cdots \ (A_B^T)^{(\kappa-1)} S_o], \quad (3.9)$$

$$\Psi_\kappa(d/dt) = [I \ d/dt \ \cdots \ I \ d^{\kappa-1}/dt^{\kappa-1}]^T. \quad (3.10)$$

Let us note that  $C_n(S_o)$  is the controllability matrix of the pair  $(A_B^T, S_o)$ . We do the following assumptions:

*Hypothesis 5.* The subspace  $A_B^T \text{Im } S_o$  is contained in the unobservable subspace  $\cap_{i=1}^n \ker C_B (A_B^T)^{i-1}$ , namely:

$$C_B A_B^T C_n(S_o) = 0. \quad (3.11)$$

*Hypothesis 6.* The state space description  $\Sigma(A_B, B_B, C_B)$  (2.4) has no finite invariant zeros at the origin, namely:

$$\text{Im } B_B \cap A_B \ker C_B = \{\mathbf{0}\}. \quad (3.12)$$

As we will see later, Hypothesis 5 assures the rejectibility of the non-linear part  $q_o$ , and Hypothesis 6 assures the left invertibility of the static gain of  $\Sigma_{ss}(A_B, B_B, C_B)$ .

We shall need the following two technical Lemmas:

*Lemma 7.* If (3.11) holds, then the operator  $X(d/dt)$ , defined by (3.8), satisfies:

$$\begin{bmatrix} N_B & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} (Id/dt - A_B) & -S_o \\ C_B & 0 \end{bmatrix} \begin{bmatrix} X(d/dt) \\ I_n \end{bmatrix} = 0, \quad (3.13)$$

where  $N_B$  is the maximal annihilator of  $\text{Im } B_B$ ; namely:  $N_B B_B = 0$ ,  $\ker \begin{bmatrix} N_B \\ B_B^T \end{bmatrix} = \{\mathbf{0}\}$ , and  $\text{Im} \begin{bmatrix} N_B \\ B_B^T \end{bmatrix} = \mathbb{R}^n$ .

**Proof of Lemma 7** The maximal annihilator  $N_B$  is:

$$N_B = \mathbf{BD} \{N_1, \dots, N_m\}, \quad (3.14)$$

where:  $N_i = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(n_i-1) \times n_i}$ . Then:<sup>1</sup>

$$\begin{aligned} N_B(\text{Id}/dt - A_B)X(d/dt) &= \\ = N_B(A_B A_B^T - A_B^T d/dt)C_n(S_o)\Psi_n(d/dt) &\equiv N_B S_o \end{aligned} \quad (3.15)$$

And from (3.11), we get (3.13).  $\square$

*Lemma 8.* The operator  $X(d/dt)$ , defined by (3.8), satisfies:

$$\begin{aligned} B_B^T(-\text{Id}/dt - A_B)X(d/dt) + S_o \\ = B_B^T C_{(n+1)}(S_o)\Psi_{(n+1)}(d/dt). \end{aligned} \quad (3.16)$$

Furthermore:

$$B_B^T C_{(n+1)}(S_o)\Psi_{(n+1)}(d/dt) = \Sigma_S(d/dt), \quad (3.17)$$

where:

$$\begin{aligned} \Sigma_S(d/dt) &= [\sigma_{S_1}(d/dt) \dots \sigma_{S_m}(d/dt)]^T, \\ \sigma_{S_i}^T(d/dt) &= [\sigma_{S_{i,1}}(d/dt) \dots \sigma_{S_{i,\ell}}(d/dt)], \\ \sigma_{S_{i,j}}(d/dt) &= \det \begin{bmatrix} (I_{n_i} d/dt - A_i) & \mathbf{S}_{i,j} \\ -\mathbf{c}_i^T & 0 \end{bmatrix}, \end{aligned} \quad (3.18)$$

$i \in \{1, \dots, m\}$ ,  $j \in \{1, \dots, \ell\}$ , and the  $\mathbf{c}_i$  are vectors in  $\ker N_i A_i$  such that:  $\mathbf{c}_i^T \mathbf{c}_i = 1$ .

**Proof of Lemma 8** Equation (3.16) directly follows from (3.8) and the fact:  $B_B^T A_B A_B^T = 0$ . Equation (3.17) directly follows from (3.18), (2.5) and (2.6).  $\square$

In the next Theorem, we introduce a change of variable which aim is to map the nonlinear perturbation signal,  $\mathbf{q}_o$ , into the image of  $B_B$ .

*Theorem 9.* Let  $\zeta$  be the state variable defined as follows:

$$\zeta = \xi - X(d/dt)\mathbf{q}_o(\xi, \bar{\mathbf{u}}). \quad (3.19)$$

where  $X(d/dt)$  is the operator defined by (3.8). Then the state representation (2.4) takes the following form:

$$d\zeta/dt = A_B \zeta + B_B(\bar{\mathbf{u}} + \mathbf{q}_*(\xi, \bar{\mathbf{u}})); \mathbf{y}(t) = C_B \zeta(t), \quad (3.20)$$

where:  $\mathbf{q}_*(\xi, \bar{\mathbf{u}}) = \Sigma_S(d/dt)\mathbf{q}_o(\xi, \bar{\mathbf{u}})$ .

**Proof of Theorem 9** Substituting (3.19) into (2.4), we get from Lemmas 7 and 8:  $N_B(\text{Id}/dt - A_B)\zeta = 0$ ,  $C_B \zeta = \mathbf{y}$ , and:<sup>2</sup>

$$\begin{aligned} B_B^T(\text{Id}/dt - A_B)\zeta &= \\ B_B^T(\text{Id}/dt - A_B)(\xi - X(d/dt)\mathbf{q}_o(\xi, \bar{\mathbf{u}})) &= \\ B_B^T(\text{Id}/dt - A_B)\xi + (\Sigma_S(d/dt) - B_B^T S_o) &\mathbf{q}_o(\xi, \bar{\mathbf{u}}) = \\ = \bar{\mathbf{u}} + \Sigma_S(d/dt)\mathbf{q}_o(\xi, \bar{\mathbf{u}}). \end{aligned} \quad \square$$

Thus, with the help of the change of variable (3.19), we have carried the state representation (2.4) to (3.20), which has the particularity that the nonlinearities  $\mathbf{q}_*(\xi, \bar{\mathbf{u}})$  are contained in  $\text{Im } B_B$ . Then, with the ideal state feedback:

$$\bar{\mathbf{u}}_* = F\zeta - \mathbf{q}_*(\xi, \bar{\mathbf{u}}), \quad (3.21)$$

we get an exact feedback linearization, where  $F \in \mathbb{R}^{n \times m}$  is any stabilizing feedback of the pair  $(A_B, B_B)$  namely:<sup>3</sup>

$$\sigma\{(A_B + B_B F)\} \subset \mathbb{C}^-. \quad (3.22)$$

<sup>1</sup> Let us note that (recall (2.6), (3.14) and (3.8)-(3.10)): (i)  $N_B A_B A_B^T = N_B$ , (ii)  $(A_B^T)^n = 0$  and (iii)  $C_n \Psi_n(d/dt) = (I + (A_B^T)^n d/dt + \dots + (A_B^T)^{n-1} d^{n-1}/dt^{n-1}) S_o$ .

<sup>2</sup> Recall (3.19) (3.16), (3.17) and (2.4), and that:  $B_B^T B_B = I$ .

<sup>3</sup> Given a square matrix  $A$ ,  $\sigma\{A\}$  stands for its spectrum.

#### 4. STRUCTURAL ASYMPTOTIC FEEDBACK LINEARIZATION

Remember that the exact feedback linearization (3.21) requires a complete knowledge of the model, and particularly the time derivatives of the nonlinear variable  $\mathbf{q}_o(\xi, \bar{\mathbf{u}})$ . But this approach has the advantage, over the previous classical feedback linearization, that the unknown nonlinearities  $\mathbf{q}_*(\xi, \bar{\mathbf{u}})$  belong to  $\text{Im } B_B$ . So, they can be asymptotically attenuated with the help of disturbance rejectors based on standard state observers. For example, Bonilla *et al* (2016) have proposed the following disturbance rejector based on the Beard-Jones filter (*cf.* Isermann (1984); Massoumnia (1986); Saberi *et al* (2000)):

$$\begin{aligned} d\mathbf{w}/dt &= (A_B + K C_B)\mathbf{w} - K\mathbf{y} + B_B \bar{\mathbf{u}}, \\ \bar{\mathbf{r}} &= G^\ell (C_B \mathbf{w} - \mathbf{y}), \quad \bar{\mathbf{u}} = F\zeta + \bar{\mathbf{r}}, \end{aligned} \quad (4.23)$$

where  $K \in \mathbb{R}^{\hat{n} \times p}$  is an output injection to be computed, such that:

$$\sigma\{(A_B + K C_B)\} \subset \mathbb{C}^-. \quad (4.24)$$

And  $G^\ell$  is a left inverse of the static gain,  $-C_B A_{B_K}^{-1} B_B$ , of the remainder generator:<sup>4</sup>

$$d\bar{\mathbf{e}}/dt = A_{B_K} \bar{\mathbf{e}} - B_B \mathbf{q}_*(\xi, \bar{\mathbf{u}}); \mathbf{r} = C_B \bar{\mathbf{e}}, \quad (4.25)$$

where:  $A_{B_K} \doteq (A_B + K C_B)$ ,  $\mathbf{e}(t) = \mathbf{w}(t) - \zeta(t)$  and  $\mathbf{r} = C_B \mathbf{w} - \mathbf{y}$ . Let us note that:<sup>5</sup>  $\bar{\mathbf{r}}(s) = -G^\ell F(s)\mathbf{q}_*(s)$ , where:  $F(s) \doteq C_B (sI - A_{B_K})^{-1} B_B$ . Then, under the assumption that  $\mathbf{q}_*(\xi, \bar{\mathbf{u}})$  is a bounded limited band frequency signal<sup>6</sup>, we only need to synthesize a Hurwitz low-pass filter  $F(s)$ , with a corner frequency  $\omega_c$ , to achieve an asymptotic feedback linearization, indeed:

$$\|\bar{\mathbf{u}}_*(\omega) - \bar{\mathbf{u}}(\omega)\| \leq \|G^\ell F(j\omega) - I\| \|\mathbf{q}_*(\omega)\|. \quad (4.26)$$

In (Bonilla *et al*, 2016) is shown (see their Lemma 1 and Theorem 1):

Under the Hurwitz stability assumptions (4.24) and (3.22), there exist  $k_3, k_4, \alpha_2 \in \mathbb{R}^+$  such that the closed loop system behaves as:<sup>7</sup>

$$\left| \mathbf{y}(t) - C_B \int_0^t \exp(A_{B_F}(t-\tau)) B_B \mathbf{u}_r(\tau) d\tau \right| \leq (k_3/\omega_c) \|d\mathbf{q}_*(t)/dt\|_\infty + k_4 e^{-\alpha_2 t},$$

where:  $A_{B_F} \doteq (A_B + B_B F)$ . Moreover, the steady state response of  $y$  for high corner frequencies is:

$$\mathbf{y}_{ssr} = \lim_{\substack{t \rightarrow \infty \\ \omega_c \rightarrow \infty}} \mathbf{y}(t) = C_B \int_0^\infty \exp(A_{B_F}(t-\tau)) B_B \mathbf{u}_r(\tau) d\tau$$

#### 5. QUADROTOR

Let us consider a Quadrotor where the total mass is  $M$ , the moment of inertia about axis  $ox$ ,  $oy$  and  $oz$  are:<sup>8</sup>  $\mathcal{I}_{xx}$ ,

<sup>4</sup> Recall (3.12), and note that:  $(A_B + K C_B) \ker C_B = A_B \ker C_B$ .

<sup>5</sup> In order to clarify ideas, let us assume for a while the existence of the Laplace transform of  $\mathbf{q}_*(\xi, \bar{\mathbf{u}})$ .

<sup>6</sup> Since  $F$  is a stabilizing feedback of the pair  $(A_B, B_B)$ , this assumption is satisfied in a local neighborhood around the equilibrium point  $(\xi, \bar{\mathbf{u}}) = (\mathbf{0}, \mathbf{0})$ .

<sup>7</sup>  $\mathbf{u}_r$  is some reference signal added to the control law (4.23).

<sup>8</sup> Since the Quadrotor is mechanically symmetric its products inertia are zero.

$\mathcal{I}_{yy}$  and  $\mathcal{I}_{zz}$ , and the distance of each rotor with respect to the centre of gravity of the quadrotor is:  $L$  (see Fig. 1).

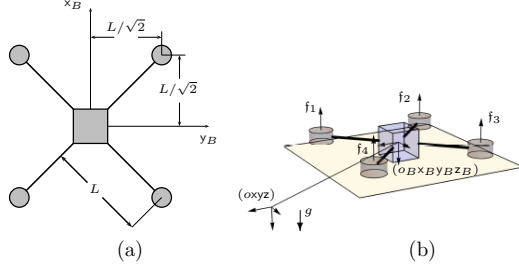


Fig. 1. Schematic diagram of the quadrotor. (a) Upper view. (b) Perspective view.

The motion is referred to a fixed orthogonal axis set (airframe)  $(oxyz)$ , where  $oz$  points vertically down along the gravity vector  $[0 \ 0 \ g]^T$ , and the origin  $o$  is located at the desired height  $\bar{z}$ , above the ground level.  $\phi$ ,  $\theta$  and  $\psi$  are the Euler angles, roll, pitch and yaw, measured respectively over the axis  $o_B \times o_B$ ,  $o_B \times o_B$  and  $o_B \times o_B$ ; where  $(o_B \times o_B \times o_B \times o_B)$  is the body axis system, with its origin  $o_B$  fixed at the centre of gravity of the quadrotor. We denote:  $\eta = [\phi \ \theta \ \psi]^T$ . Usually, the Quadrotor is represented by the following behavioral equations (see for example García *et al* (2013) and Cook (2013)):

**Translational dynamics** The translational behavior is represented by the non-linear state description:

$$\begin{aligned} \ddot{x} &= -\theta g + q_x (\Delta u_z; \eta), & \ddot{y} &= \phi g + q_y (\Delta u_z; \eta), \\ \ddot{z} &= \Delta u_z / M + q_z (\Delta u_z; \eta), \end{aligned} \quad (5.27)$$

where:  $\ddot{x} \doteq d^2x/dt^2$ ,  $\ddot{y} \doteq d^2y/dt^2$ ,  $\ddot{z} \doteq d^2z/dt^2$ , and  $q_x$ ,  $q_y$  and  $q_z$  are the variables describing the non-linear part of the translational model (see the Appendix), and  $\Delta u_z$  is the incremental control action:  $\Delta u_z = u_z + Mg$ , and  $u_z$  is defined hereafter in the thrusters model.

**Attitude dynamics** The attitude behavior is represented by the non-linear state description:

$$\begin{aligned} \ddot{\phi} &= u_y / \mathcal{I}_{xx} + q_\phi (\eta), & \ddot{\theta} &= u_x / \mathcal{I}_{yy} + q_\theta (\eta), \\ \ddot{\psi} &= u_\psi / \mathcal{I}_{zz} + q_\psi (\eta), \end{aligned} \quad (5.28)$$

where:  $\ddot{\phi} \doteq d^2\phi/dt^2$ ,  $\ddot{\theta} \doteq d^2\theta/dt^2$ ,  $\ddot{\psi} \doteq d^2\psi/dt^2$ , and  $q_\phi$ ,  $q_\theta$  and  $q_\psi$  are the variables describing the non-linear part of the attitude model (see the Appendix), and  $u_y$ ,  $u_x$  and  $u_\psi$  are the incremental control actions:  $u_y = \tau_\phi$ ,  $u_x = \tau_\theta$  and  $u_\psi = \tau_\psi$ , defined hereafter.

**Thrusters model** The control actions,  $u_z$ ,  $\tau_\phi$ ,  $\tau_\theta$  and  $\tau_\psi$ , are related with the thrusters of the four rotors,  $f_1$ ,  $f_2$ ,  $f_3$  and  $f_4$ , by means of the following invertible matrix:

$$\begin{bmatrix} u_z \\ u_y \\ u_x \\ u_\psi \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ k_\ell & k_\ell & -k_\ell & -k_\ell \\ -k_\ell & k_\ell & k_\ell & -k_\ell \\ -\gamma & \gamma & -\gamma & \gamma \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

where:  $k_\ell = L/\sqrt{2}$  and  $\gamma = k_\tau/k_f$ <sup>9</sup> (the determinant of this matrix is:  $-16\gamma k_\ell^2$ ).

<sup>9</sup> The thrusters and momentums generated by each rotor are:  $f_i = k_f \omega_i^2$  and  $\tau_i = k_\tau \omega_i^2$ , where  $\omega_i$  is the angular velocity of each rotor. Thus:  $\tau_i = (k_\tau/k_f) f_i$ , which implies (Powers *et al*, 2014):  $\gamma = k_\tau/k_f$ .

### 5.1 State Representations

From (5.27) and (5.28), we get the following state representations (cf. (2.3) and (2.4)):

**State representation of x-dynamics :**

$$d\xi_x/dt = A_{Bx}\xi_x + B_{Bx}u_x + S_{ox}q_{ox}; \quad y_x = C_{Bx}\xi_x, \quad (5.29)$$

$$A_{Bx} = C_4 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B_{Bx} = \chi_4^4, \quad C_{Bx}^T = (-g/\mathcal{I}_{yy}) \chi_4^1, \quad (5.30)$$

$$S_{ox} = [(-\mathcal{I}_{yy}/g) \chi_4^2 \quad (\mathcal{I}_{yy}) \chi_4^4],$$

where:  $\xi_x = T_{Bx}^{-1}x_x$ ,  $x_x = [x \ dx/dt \ \theta \ d\theta/dt]^T$ ,  $T_{Bx} = \mathbf{BD} \{-g/\mathcal{I}_{yy}, -g/\mathcal{I}_{yy}, 1/\mathcal{I}_{yy}, 1/\mathcal{I}_{yy}\}$ ,  $q_{ox} = [q_x \ q_\theta]^T$ ,  $A_{Bx} = T_{Bx}^{-1}A_{ox}T_{Bx}$ ,  $B_{Bx} = T_{Bx}^{-1}B_{ox}$ ,  $C_{Bx} = C_{ox}T_{Bx}$ ,  $S_{ox}q_{ox} = T_{Bx}^{-1}f_{ox}$  and  $f_{ox} = (q_x)\chi_4^2 + (q_\theta)\chi_4^4$ .

Let us note that (3.11) is satisfied:  $C_{Bx}A_{Bx}^T = 0$ .

**State representation of y-dynamics :**

$$d\xi_y/dt = A_{By}\xi_y + B_{By}u_y + S_{oy}q_{oy}; \quad y_y = C_{By}\xi_y, \quad (5.31)$$

$$A_{By} = C_4 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B_{By} = \chi_4^4, \quad C_{By}^T = (g/\mathcal{I}_{xx}) \chi_4^1, \quad (5.32)$$

$$S_{oy} = [(\mathcal{I}_{xx}/g) \chi_4^2 \quad (\mathcal{I}_{xx}) \chi_4^4],$$

where:  $\xi_y = T_{By}^{-1}x_y$ ,  $x_y = [y \ dy/dt \ \phi \ d\phi/dt]^T$ ,  $T_{By} = \mathbf{BD} \{g/\mathcal{I}_{xx}, g/\mathcal{I}_{xx}, 1/\mathcal{I}_{xx}, 1/\mathcal{I}_{xx}\}$ ,  $q_{oy} = [q_y \ q_\phi]^T$ ,  $A_{By} = T_{By}^{-1}A_{oy}T_{By}$ ,  $B_{By} = T_{By}^{-1}B_{oy}$ ,  $C_{By} = C_{oy}T_{By}$ ,  $S_{oy}q_{oy} = T_{By}^{-1}f_{oy}$  and  $f_{oy} = (q_y)\chi_4^2 + (q_\phi)\chi_4^4$ .

Let us note that (3.11) is satisfied:  $C_{By}A_{By}^T = 0$ .

**State representation of ψ-dynamics :**

$$dx_\psi/dt = A_{o\psi}x_\psi + B_{o\psi}u_\psi + f_{o\psi}; \quad \psi = C_{o\psi}x_\psi$$

$$A_{o\psi} = C_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B_{o\psi} = (\mathcal{I}_{zz}^{-1}) \chi_2^2, \quad f_{o\psi} = (q_\psi) \chi_2^2, \quad (5.33)$$

$$C_{o\psi}^T = \chi_2^1,$$

where:  $x_\psi = [\psi \ d\psi/dt]^T$ .

**State representation of z-dynamics :**

$$dx_z/dt = A_{oz}x_z + B_{oz}\Delta u_z + f_{oz}; \quad z = C_{oz}x_z$$

$$A_{oz} = C_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B_{oz} = (M^{-1}) \chi_2^2, \quad f_{oz} = (q_z) \chi_2^2, \quad (5.34)$$

$$C_{oz}^T = \chi_2^1,$$

where:  $x_z = [z \ dz/dt]^T$ .

The state representations (5.33) and (5.34) are already in the form (3.20) of Theorem 9.

### 5.2 Structural Decomposition

In order to get the state representation (3.20) of Theorem 9, we need the operator (cf. (3.8)-(3.10)):

$$X_x(d/dt) = -A_{Bx}^T C_4 (S_{ox}) \Psi_4(d/dt) = \frac{\mathcal{I}_{yy}}{g} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ d/dt & 0 \end{bmatrix}$$

Thus:

$$C_{Bx}X_x(d/dt) \equiv 0,$$

$$S_{ox} - (I_4 d/dt - A_{Bx}) X_x(d/dt) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -((\mathcal{I}_{yy}/g)d^2/dt^2) \mathcal{I}_{yy} \end{bmatrix},$$

namely:

$$q_{*,x} = -(\mathcal{I}_{yy}/g) d^2 q_x/dt^2 + \mathcal{I}_{yy} q_\theta. \quad (5.35)$$

In the same way:

$$q_{*,y} = (\mathcal{I}_{xx}/g) d^2 q_y/dt^2 + \mathcal{I}_{xx} q_\phi. \quad (5.36)$$

## 6. NUMERICAL SIMULATIONS

We have considered the following numerical values, taken from a laboratory prototype:

$$M = 0.60 \text{ kg}, \mathcal{I}_{xx} = 0.00, 32 \text{ kg m}^2, \mathcal{I}_{yy} = 0.00, 32 \text{ kg m}^2, \\ \mathcal{I}_{zz} = 0.00, 58 \text{ kg m}^2, L = 0.17 \text{ m}, g = 9.81 \text{ m s}^{-2}.$$

The control laws are the following (*cf.* (4.23)):

$$\begin{aligned} \text{x-dynamics} \quad &: \\ \frac{d\mathbf{w}_x}{dt} &= \mathbf{A}_{\bar{K}_x} \mathbf{w}_x - \bar{K}_x \mathbf{x} + \bar{B}_{Bx} u_x, \\ u_x &= F_x \zeta_x + \bar{r}_x, \quad \bar{r}_x = \bar{G}_x^\ell (\bar{C}_{Bx} \mathbf{w}_x - \mathbf{x}), \end{aligned} \quad (6.37)$$

where:  $k_x = -g \mathcal{I}_{yy}^{-1}$ ,  $\bar{B}_{Bx} = k_x B_{Bx}$ ,  $\bar{C}_{Bx} = k_x^{-1} C_{Bx}$ ,  $\mathbf{A}_{\bar{K}_x} = (A_{Bx} + \bar{K}_x \bar{C}_{Bx})$ ,  $\bar{G}_x^\ell = -\bar{C}_{Bx} \mathbf{A}_{\bar{K}_x}^{-1} \bar{B}_{Bx}$ ,  $\zeta_x = [x \, dx/dt \, d^2x/dt^2 \, d^3x/dt^3]^T$ ,  $\zeta_x = k_x^{-1} \bar{\zeta}_x$ .

$$\begin{aligned} \text{y-dynamics} \quad &: \\ \frac{d\mathbf{w}_y}{dt} &= \mathbf{A}_{\bar{K}_y} \mathbf{w}_y - \bar{K}_y \mathbf{y} + \bar{B}_{By} u_y, \\ u_y &= F_y \zeta_y + \bar{r}_y, \quad \bar{r}_y = \bar{G}_y^\ell (\bar{C}_{By} \mathbf{w}_y - \mathbf{y}), \end{aligned} \quad (6.38)$$

where:  $k_y = g \mathcal{I}_{xx}^{-1}$ ,  $\bar{B}_{By} = k_y B_{By}$ ,  $\bar{C}_{By} = k_y^{-1} C_{By}$ ,  $\mathbf{A}_{\bar{K}_y} = (A_{By} + \bar{K}_y \bar{C}_{By})$ ,  $\bar{G}_y^\ell = -\bar{C}_{By} \mathbf{A}_{\bar{K}_y}^{-1} \bar{B}_{By}$ ,  $\zeta_y = [y \, dy/dt \, d^2y/dt^2 \, d^3y/dt^3]^T$ ,  $\zeta_y = k_y^{-1} \bar{\zeta}_y$ .

$$\begin{aligned} \text{\(\psi\)-dynamics} \quad &: \\ \frac{d\mathbf{w}_\psi}{dt} &= \mathbf{A}_{\bar{K}_\psi} \mathbf{w}_\psi - \bar{K}_\psi \psi + \bar{B}_{B\psi} u_\psi, \\ u_\psi &= F_\psi \zeta_\psi + \bar{r}_\psi, \quad \bar{r}_\psi = \bar{G}_\psi^\ell (\bar{C}_{B\psi} \mathbf{w}_\psi - \psi), \end{aligned} \quad (6.39)$$

where:  $A_{B\psi} = A_{o\psi}$ ,  $\bar{B}_{B\psi} = B_{o\psi}$ ,  $\bar{C}_{B\psi} = C_{o\psi}$ ,  $\mathbf{A}_{\bar{K}_\psi} = (A_{B\psi} + \bar{K}_\psi \bar{C}_{B\psi})$ ,  $\bar{G}_\psi^\ell = -\bar{C}_{B\psi} \mathbf{A}_{\bar{K}_\psi}^{-1} \bar{B}_{B\psi}$ ,  $\zeta_\psi = [\psi \, d\psi/dt]^T$ ,  $\zeta_\psi = k_\psi^{-1} \bar{\zeta}_\psi$ ,  $k_\psi = \mathcal{I}_{zz}^{-1}$ .

**z-dynamics** From (5.34), we have that:  $\bar{r}_z = -M q_z$ . And from (A.1(c)), it follows that:  $\bar{r}_z = M g \bar{q}_z / (1 + \bar{q}_z)$ , then:

$$\Delta u_z = F_z \zeta_z + \bar{r}_z, \quad \bar{r}_z = k_z^{-1} g \bar{q}_z / (1 + \bar{q}_z), \quad (6.40)$$

where:  $\bar{\zeta}_z = \mathbf{z}_z = [z \, dz/dt]^T$ ,  $\zeta_z = k_z^{-1} \bar{\zeta}_z$ ,  $k_z = M^{-1}$ .

The state feedbacks,  $F_x$ ,  $F_y$ ,  $F_\psi$  and  $F_z$ , were computed to minimize the criterions:

$$J_i = \int_0^\infty (\zeta_i^T \zeta_i + (1/5) u_i^2) dt, \quad i \in \{x, y, \psi, z\},$$

obtaining:

$$F_x = F_y = [-0.4472 \quad -1.5814 \quad -2.5725 \quad -2.3119], \\ F_\psi = F_z = [-0.4472 \quad -1.0461],$$

with the following dynamics (spectrums):<sup>10</sup>

$$\Lambda_{A_{F_i}}(s) = \{-0.4162 \pm 0.7387 i, -0.7398 \pm 0.2736 i\}, \\ \Lambda_{A_{F_j}}(s) = \{-0.5231 \pm 0.4167 i\}, \quad i \in \{x, y\}, \quad j \in \{\psi, z\}.$$

<sup>10</sup>  $\Lambda_{F_k} = (A_{Bk} + B_{Bk} F_k)$ ,  $k = i, j$ .

The output injections,  $\bar{K}_x$ ,  $\bar{K}_y$ ,  $\bar{K}_\psi$  and  $\bar{K}_z$ , were computed for having a second and a fourth order approximations of the Gaussian filter ( $\kappa = 2, 4$ ):  $H(s)H(-s) = 1 / \sum_{i=0}^{\kappa} \frac{(-1)^i}{i!} s^{2i}$  (see for example Blinichikoff and Zverev (1976)), with the spectral radii,  $\rho_K$ , of the  $(A_{B_i} + \bar{K}_i \bar{C}_{B_i})$  100 times greater than the ones,  $\rho_F$ , of the  $(A_{B_i} + B_{B_i} F_i)$ ,  $i \in \{x, y, \psi\}$ , obtaining:

$$\bar{K}_i = [-430.1 \quad -78, 122 \quad -6.962 \times 10^6 \quad -2.532 \times 10^8], \\ i \in \{x, y\}, \quad \bar{K}_\psi = [-146.9 \quad -6, 325].$$

with the following spectra:

$$\Lambda_{A_{\bar{K}_i}}(s) = \{-100.1 \pm 89.9 i, -114.9 \pm 27.81 i\}, \\ \Lambda_{A_{\bar{K}_j}}(s) = \{-73.47 \pm 30.43 i\}, \quad i \in \{x, y\}, \quad j \in \{\psi, z\}.$$

Let us show some simulation results obtained in a MATLAB<sup>®</sup> platform. We have considered that the airframe (*oxyz*) is located at height  $\bar{z} = 0.3$  [m], above the ground level, and with the initial conditions:  $\mathbf{x}(0) = \mathbf{y}(0) = \mathbf{z}(0) = \mathbf{0}$  [m],  $dx(0)/dt = dy(0)/dt = dz(0)/dt = 0$  [m s<sup>-1</sup>],  $\phi(0) = \theta(0) = \psi(0) = \alpha_0$ ,  $\alpha_0 = \pi/6$  [rad],  $d\phi(0)/dt = d\theta(0)/dt = d\psi(0)/dt = 0$  [rad s<sup>-1</sup>].

The initial conditions of the Beard-Jones filters (6.37(a)), (6.38(a)) and (6.39(a)) were set up as (see the Appendix):

$$\mathbf{w}_x(0) = [0 \quad 0 \quad -g \tan \alpha_0 (\cos \alpha_0 + \tan \alpha_0) \quad 0]^T, \\ \mathbf{w}_y(0) = [0 \quad 0 \quad g \tan \alpha_0 (1 - \sin \alpha_0) \quad 0]^T, \quad \mathbf{w}_\psi(0) = [\alpha_0 \quad 0]^T.$$

In Fig. A.1, we show the simulation results.

## 7. CONCLUSION

In this paper, we have proposed a structural feedback linearization based on failure detection techniques. We have considered a non-linear system, represented by a non-linear state representation (1.1), for which there exists a structural differential operator  $X(d/dt)$ , (3.8)-(3.8), which transforms (1.1) into the state description (3.20), where the non-linear components  $\mathbf{q}_*$  are contained in the image of its constant input matrix  $B_B$ . The operator  $X(d/dt)$  depends on the way the constant matrix,  $S_o$ , of the non-linear components,  $\mathbf{q}_o$ , is related with the internal structure of the linear part, defined by the pair  $(A_o, B_o)$ , *cf.* (1.2).

Once arrived to (3.20), we only need to directly cancel  $\mathbf{q}_*$ , if it is of course known. And if this is not the case, it can be simply asymptotically rejected with a failure reconstructor, as for example, the one proposed in (Bonilla *et al.*, 2016).

One advantage of this control schema is that it enlarges the linearity neighborhood around the equilibrium point  $\mathbf{0}$ . Also, the possible uncertainty's model is absorbed by  $\mathbf{q}_*$ .

We have used a quadrotor example to illustrate this performant alternative technique. In the simulation results, we have obtained a good control performance until a perturbation of  $\pi/4$  rad in the Euler angles  $(\phi, \theta, \psi)$ ; here is reported the case  $\pi/3$  rad.

Preprints of the 20th IFAC World Congress  
Toulouse, France, July 9-14, 2017

## REFERENCES

- Beard, R.V. (1971). Failure accommodation in linear systems through self-reorganization. (*PhD thesis, Massachusetts Institute of Technology, 1971.*)
- Blinchikoff H.J., and Zverev, A.I. (1976). **Filtering in the Time and Frequency Domains**. New York: John Wiley & Sons, Inc.
- Bonilla M., L.A. Blas, S. Salazar, J.C. Martínez and M. Malabre (2016). A Robust Linear Control Methodology based on Fictitious Failure Rejection. *European Control Conference*, pp. 2596–2601. June 29 - July 1, 2016. Aalborg, Denmark.
- Brunovsky, P. (1970). A classification of linear controllable systems. *Kybernetika* **6**(3), 173–188.
- Cook M.V. (2013). **Flight Dynamics Principles. A Linear Systems Approach to Aircraft Stability and Control Elsevier Ltd., New York.**
- García Carrillo L.R., A.E. Dzul López, R. Lozano and C. Pégard (2013). **Quad Rotorcraft Control. Vision-Based Hovering and Navigation Springer-Verlag, London.**
- Isermann, R. (1984). Process fault detection based on modeling and estimation methods-A survey, *Automatica*, **20**, 387–404.
- Isidori A. (1989). **Nonlinear Control Systems, An Introduction, Springer-Verlag, Berlin.**
- Khalil H.K. (1992). **Nonlinear Systems, Macmillan Publishing Company New York.**
- Massoumnia. M.-A. (1986). A geometric approach to the synthesis of failure detection filters. *IEEE Trans. Automatic Control*, 31(9), 1986, 839-846.
- Nijmeijer H. and A.J. Van der Schaft (1990). **Nonlinear dynamical control systems, New York : Springer-Verlag.**
- Powers C., D. Mellinger, and V. Kumar (2014). Quadrotor Kinematics and Dynamics. In Chapter 16 of **Handbook of unmanned aerial vehicles**, editors Valavanis, K. P., & Vachtsevanos, G. J. New York : Springer Publishing Company, Incorporated.
- Slotine, JJ E and Li W. (1991). **Applied Nonlinear Control, Prentice Hall NY.**
- Saberi, A., Stoorvogel, A.A., Sannuti, P. & Niemann, H.H. (2000). Fundamental problems in fault detection and identification. *International Journal of Robust and Nonlinear Control*, **10**, 1209–1236.
- Vidyasagar M. (1993). **Nonlinear Systems Analysis, 2nd edition Prentice-Hall, Inc. New Jersey.**

## Appendix A. VARIABLES DEFINITIONS

In this Appendix, we use the abbreviated notations:  $(c_\phi, c_\theta, c_\psi)$  for  $(\cos \phi, \cos \theta, \cos \psi)$ ,  $(s_\phi, s_\theta, s_\psi)$  for  $(\sin \phi, \sin \theta, \sin \psi)$  and  $(\dot{\phi}, \dot{\theta}, \dot{\psi})$  for  $(d\phi/dt, d\theta/dt, d\psi/dt)$ . We also omit the explicit dependence,  $(\Delta u_z; \eta)$ , from the variables definitions,  $q_x, q_y$  and  $q_z$ .

The variables  $q_x, q_y$  and  $q_z$ , describing the non-linear part are:

$$q_x = \theta g + (\Delta u_z/M - g) \bar{q}_x, \quad q_y = -\phi g + (\Delta u_z/M - g) \bar{q}_y, \quad (A.1)$$

$$q_z = (\Delta u_z/M - g) \bar{q}_z,$$

$$\bar{q}_x = c_\phi s_\theta c_\psi + s_\phi s_\psi, \quad \bar{q}_y = c_\phi s_\theta s_\psi - s_\phi c_\psi, \quad \bar{q}_z = c_\phi c_\theta - 1. \quad (A.2)$$

Note that:  $q_x(\Delta u_z; 0) = 0$ ,  $q_y(\Delta u_z; 0) = 0$  and  $q_z(\Delta u_z; 0) = 0$ .

The variables  $q_\phi, q_\theta$  and  $q_\psi$ , describing the non-linear part of the attitude model, are defined as follows  $(\mathbf{q}_\eta = [q_\phi \ q_\theta \ q_\psi]^T)$ :

$$\mathbf{q}_\eta = (J^{-1}(\eta) - J^{-1}(0))\tau - J^{-1}(\eta)C(\eta, \dot{\eta})\dot{\eta} \quad (A.3)$$

The elements of the symmetric inertial matrix  $J$  are:

$$J_{11} = I_{xx}, \quad J_{22} = I_{yy}c_\phi^2 + I_{zz}s_\phi^2, \quad J_{23} = (I_{yy} - I_{zz})c_\phi s_\phi c_\theta,$$

$$J_{12} = 0, \quad J_{13} = -I_{xx}s_\theta, \quad J_{33} = I_{xx}s_\theta^2 + I_{yy} - I_{zz}c_\phi s_\phi c_\theta$$

The elements of the Coriolis matrix  $C(\eta, \dot{\eta})$  are  $(\Delta_{zy} = (I_{zz} - I_{yy}))$ :

$$c_{11} = 0, \quad c_{12} = -I_{xx}c_\theta\dot{\psi} - \Delta_{zy}(c_\phi s_\phi\dot{\theta} + (s_\phi^2 - c_\phi^2)c_\theta\dot{\psi}),$$

$$c_{13} = \Delta_{zy}c_\phi s_\phi c_\theta^2\dot{\psi}, \quad c_{21} = I_{xx}c_\theta\dot{\psi} + \Delta_{zy}(c_\phi s_\phi\dot{\theta} + (s_\phi^2 - c_\phi^2)c_\theta\dot{\psi}),$$

$$c_{22} = \Delta_{zy}c_\phi s_\phi\dot{\phi}, \quad c_{23} = (I_{zz}c_\phi^2 + I_{yy}s_\phi^2 - I_{xx})c_\theta s_\theta\dot{\psi},$$

$$c_{31} = -I_{xx}c_\theta\dot{\theta} - \Delta_{zy}c_\phi s_\phi c_\theta^2\dot{\psi}, \quad c_{32} = I_{xx}c_\theta s_\theta\dot{\psi} +$$

$$I_{yy}((c_\phi^2 - s_\phi^2)c_\theta\dot{\phi} - c_\phi s_\phi s_\theta\dot{\theta} - s_\phi^2 c_\theta s_\theta\dot{\psi}) -$$

$$I_{zz}((c_\phi^2 - s_\phi^2)c_\theta\dot{\phi} - c_\phi s_\phi s_\theta\dot{\theta} + c_\phi^2 c_\theta s_\theta\dot{\psi}),$$

$$c_{33} = I_{xx}c_\theta s_\theta\dot{\theta} + I_{yy}(c_\phi s_\phi c_\theta\dot{\phi} - s_\phi^2 s_\theta\dot{\theta})c_\theta -$$

$$I_{zz}(s_\phi c_\theta\dot{\phi} + c_\phi c_\theta s_\theta\dot{\theta})c_\theta$$

Note that:  $C(0, \dot{\eta})\dot{\eta} = \begin{bmatrix} -(I_{xx} + I_{yy} - I_{zz})\dot{\theta}\dot{\psi} \\ (I_{xx} - I_{yy} + I_{zz})\dot{\phi}\dot{\psi} \\ -(I_{xx} - I_{yy} + I_{zz})\dot{\phi}\dot{\theta} \end{bmatrix}$ .

From (5.27(a)) and (A.1(a)), we get:  $d^2x(0)/dt^2 = -g\theta(0) + q_x(0) = (\Delta u_x(0)/M - g)\bar{q}_x(\eta(0))$ . From (6.40) and (A.2(c)), we get  $(\zeta_z(0) = 0)$ :  $\Delta u_z(0)/M - g = -g/(c_\phi(0)c_\theta(0))$ . Then:  $d^2x(0)/dt^2 = -g(s_\theta(0)c_\theta(0)^{-1}c_\psi(0) - s_\phi(0)c_\phi(0)^{-1}c_\theta(0)^{-1}s_\psi(0))$  (see (A.2(a))). In a similar way:  $d^2y(0)/dt^2 = -g(s_\theta(0)c_\theta(0)^{-1}s_\psi(0) - s_\phi(0)c_\phi(0)^{-1}c_\theta(0)^{-1}c_\psi(0))$ . Since:  $\dot{\phi}(0) = \dot{\theta}(0) = \dot{\psi}(0) = 0$ , then:  $d^3x(0)/dt^3 = d^3y(0)/dt^3 = 0$ .

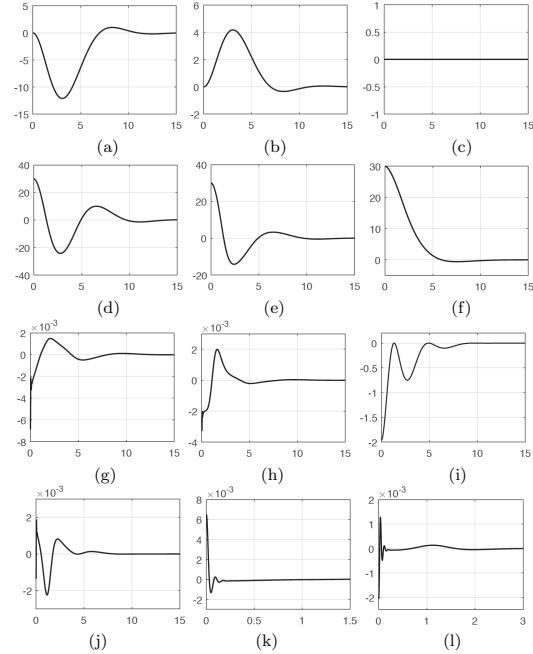


Fig. A.1. Center of mass position: (a)  $x$  [m], (b)  $y$  [m], (c)  $z$  [m]. Quadrotor attitude: (d)  $\theta$  [°], (e)  $\phi$  [°], (f)  $\psi$  [°]. Control signals: (g)  $u_x$ , (h)  $u_y$ , (i)  $\Delta u_z$ , (j)  $u_\psi$ .  $\mathbf{q}_*$  estimation: (k)  $\hat{q}_x = q_{*,x} - \bar{r}_x$ , (l)  $\hat{q}_y = q_{*,y} - \bar{r}_y$ .

## A Robust Linear Control Methodology based on Fictitious Failure Rejection

M. Bonilla, L.A. Blas, S. Salazar, J.C. Martínez and M. Malabre

**Abstract**—We propose here a quite simple and effective robust linear control scheme based on failure detection techniques. Our proposed linear control approach is intended to reject linearly structured uncertainties, the uncertainty is then modeled via a deterministic signal, that is identified as an actuator fault. The proposed robust control methodology is illustrated via the altitude control of a planar vertical takeoff and landing (PVTOL) aircraft in a laboratory setting.

### I. INTRODUCTION

There is presently a great interest in the study of the so-called “drones”, “quadri-rotors”, and “PVTOL”, aerial vehicles, among practitioners of mathematical control theory. These simple aircrafts are normally modeled as second order systems with only the inertial term, namely as simple double integrators. In practice, the vertical motion presents some nonlinear proportional and derivative terms, depending on the flying altitude. These un-modeled terms make difficult the altitude control. In [3] the authors present an altitude control scheme based on position and velocity error dynamics. Although the resulting performance is quite good, the paper does not present how to proceed to calculate the error dynamics in order to stabilize the altitude variable. In [7] the authors present an altitude control law that uses the accelerations and a compensating factor in order to improve the performance of the altitude control. Many authors, *e.g.* [6], present robust altitude control based on sliding mode controllers. However, the selection of the controllers gains is difficult, and the trimming and chattering effects are undesirable. In [9] the altitude dynamics is modeled considering the drag forces and this term is then included as a derivative effect, but in the control strategy this term is in fact neglected. The authors in [12] consider the altitude dynamics as a second order system with resistance coefficient in the velocity but in the control algorithm this component is neglected.

In this paper we study the altitude control by means of continuous-time linear state descriptions, where the matrices  $A$  and  $B$  display linearly structured uncertainty. The uncertainty is then modeled via a deterministic signal, that is

M. Bonilla and S. Salazar are with CINVESTAV-IPN, CONTROL AUTOMÁTICO, UMI 3175 CINVESTAV-CNRS. A.P. 14-740. MÉXICO 07000. mbonilla@cinvestav.mx and ssalazar@ctrl.cinvestav.mx

J. C. Martínez is with CINVESTAV-IPN, CONTROL AUTOMÁTICO. A.P. 14-740. MÉXICO 07000. juancarlos.martinez-garcia@conciliencia.org

L.A. Blas is a student of CINVESTAV-IPN, CONTROL AUTOMÁTICO. Sponsored by CONACyT, México anghel.lam@hotmail.com

M. Malabre is with LUNAM Université, CNRS, IRCCyN, UMR 6597, B.P. 92101, 44321 Nantes, Cedex 03, FRANCE. Michel.Malabre@irccyn.ec-nantes.fr

identified as an actuator fault. This unknown disruptive signal is reconstructed by means of a linear failure reconstructor. A by product of the research supporting our exposition is the conception of a robust linear control scheme based on failure detection techniques. The proposed linear robust control scheme is quite simple and quite effective against un-modeled dynamics.

The paper is structured as follows. In Section II we state our control problem, which deals with the rejection of the uncertainty that characterizes the un-modeled dynamics. For this we consider a linear control law. In Section III, we solve the proposed control problem following a failure detection methodology. In Section IV, we test our linear robust control scheme by controlling the altitude of a PVTOL aircraft in a laboratory setting. We conclude our exposition in Section V.

### II. PROBLEM STATEMENT

Let us consider the following state space representation of a given continuous-time dynamical system:

$$\begin{cases} dx/dt &= Ax + Bu, \\ y &= Cx, \end{cases} \quad (1)$$

where:  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$  and  $C \in \mathbb{R}^{p \times n}$ , with  $m \leq p$ . We assume that:  $\ker B = \{0\}$ ,  $\ker C^T = \{0\}$ , and that matrices  $A$  and  $B$  are affected by the following linearly structured uncertainty:

$$A = \bar{A} + S\tilde{A} \quad \text{and} \quad B = \bar{B} + S\tilde{B}. \quad (2)$$

We assume that the matrices  $\bar{A} \in \mathbb{R}^{n \times n}$ ,  $\bar{B} \in \mathbb{R}^{n \times m}$ ,  $C$  and  $S \in \mathbb{R}^{n \times \ell}$  are known, with  $\ell \leq p$ . We also assume that the uncertain matrices  $\tilde{A} \in \mathbb{R}^{\ell \times n}$  and  $\tilde{B} \in \mathbb{R}^{\ell \times m}$  are unknown. Moreover, assume that the following geometric conditions hold:

$$\text{im } S \subset \mathcal{V}^* + \text{im } \bar{B} \quad \text{and} \quad \text{im } S\tilde{B} \subset \mathcal{V}^*, \quad (3)$$

where  $\mathcal{V}^*$  is the supremal  $(\bar{A}, \bar{B})$  invariant subspace contained in  $\ker C$ , *i.e.*,  $\sup \{ \mathcal{V} \subset \ker C \mid \bar{A}\mathcal{V} \subset \mathcal{V} + \text{im } \bar{B} \}$ , which is the limit of the non increasing geometric algorithm:  $\mathcal{V}_0 = \mathbb{R}^n$  and  $\mathcal{V}_{\mu+1} = \ker C \cap \bar{A}^{-1}(\mathcal{V}_\mu + \text{im } \bar{B})$ ,  $\mu \in \mathbb{N}$  (see [19] and [2]).

A standard procedure in the geometric approach consists in applying first a *friend* (static state feedback control law)  $F_* \in \mathbf{F}(\mathcal{V}^*) = \{ F \in \mathbb{R}^{n \times m} \mid (\bar{A} + \bar{B}F)\mathcal{V}^* \in \mathcal{V}^* \}$ , namely:

$$u = \hat{u} + F_* x, \quad (4)$$

where  $\hat{u}$  is the new control input for the external control loop (*cf.* (23)).



From (1) and (4), we get:

$$\begin{aligned} dx/dt &= (\bar{A} + \bar{B}F_*)x + \bar{B}\hat{u} + S(q + \tilde{B}\hat{u}), \\ y &= Cx, \end{aligned} \quad (5)$$

where:

$$q := (\tilde{A} + \tilde{B}F_*)x. \quad (6)$$

Let  $\Pi \in \mathbb{R}^{\hat{n} \times n}$ , where  $\hat{n} = n - \nu$ , be the canonical projection on the quotient space  $\mathbb{R}^n / \mathcal{V}^* \approx \mathbb{R}^{\hat{n}}$ . Then, there exist unique matrices,  $\hat{A}$ ,  $\hat{B}$ ,  $\hat{S}$  and  $\hat{C}$ , such that:

$$\Pi(\bar{A} + \bar{B}F_*) = \hat{A}\Pi, \quad \Pi\bar{B} = \hat{B}, \quad \Pi S = \hat{S} \text{ and } C = \hat{C}\Pi. \quad (7)$$

Let  $P_* \in \mathbb{R}^{\nu \times n}$  be the natural projection on  $\mathcal{V}^*$  along some complementary subspace of  $\mathcal{V}^*$ , let  $\bar{V}_* \in \mathbb{R}^{n \times \nu}$  be the insertion map in  $\mathcal{V}^*$  and let  $\hat{V}_* \in \mathbb{R}^{n \times \hat{n}}$  be the insertion map in  $\mathbb{R}^n / \mathcal{V}^*$ . Let us define:

$$\begin{aligned} \bar{A}_* &= \mathcal{V}^* | (\bar{A} + \bar{B}F_*) | \mathcal{V}^* = P_*(\bar{A} + \bar{B}F_*)\bar{V}_*, \\ \hat{A}_* &= P_*(\bar{A} + \bar{B}F_*)\hat{V}_*, \\ \bar{S} &= P_*S, \\ \bar{\Delta} &= (\tilde{A} + \tilde{B}F_*)\bar{V}_*, \\ \hat{\Delta} &= (\tilde{A} + \tilde{B}F_*)\hat{V}_*. \end{aligned} \quad (8)$$

where:  $\bar{A}_* \in \mathbb{R}^{\nu \times \nu}$ ,  $\hat{A}_* \in \mathbb{R}^{\nu \times \hat{n}}$ ,  $\bar{S} \in \mathbb{R}^{\nu \times \ell}$ ,  $\bar{\Delta} \in \mathbb{R}^{\ell \times \nu}$  and  $\hat{\Delta} \in \mathbb{R}^{\ell \times \hat{n}}$ .

We then have from (5), (7) and (8) (recall that  $\text{im } S\bar{B} \subset \mathcal{V}^* = \ker \Pi$  and that  $q := (\tilde{A} + \tilde{B}F_*)x$ ):

$$\begin{aligned} \begin{bmatrix} d\bar{x}/dt \\ d\hat{x}/dt \end{bmatrix} &= \begin{bmatrix} (\bar{A}_* + \bar{S}\bar{\Delta}) & (\hat{A}_* + \bar{S}\hat{\Delta}) \\ 0 & \hat{A} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \hat{x} \end{bmatrix} \\ &+ \begin{bmatrix} P_*B \\ \hat{B} \end{bmatrix} \hat{u} + \begin{bmatrix} 0 \\ \hat{S} \end{bmatrix} q, \\ y &= \begin{bmatrix} 0 & \hat{C} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \hat{x} \end{bmatrix}, \end{aligned} \quad (9)$$

where:  $\bar{x} = P_*x$  and  $\hat{x} = \Pi x$ .

*Remark 1:* In system (9), the state decomposition resulting from the application of  $F_*$  (and the related projections), gives rise to a system ruled by the state variable  $\hat{x}$ . The motion of this variable does not depend on the dynamics of what happens inside  $\mathcal{V}^*$  (which is decoupled from the system output by the action of the chosen feedback control law). However, the uncertainty acts on  $\hat{x}$  through the dynamics of  $q$ .

Since (3) implies that:  $\text{im } \hat{S} \subset \text{im } \hat{B}$ , there then exists  $\hat{Q}$  such that:

$$\hat{S} = \hat{B}\hat{Q}. \quad (10)$$

See [19], [2] for details about this standard geometric procedure.

Let us state the following problem:

*Problem 1:* (Structural uncertainty rejection)

Given (8), let us assume that:

$$\sigma\{\bar{A}_*\} \subset \mathbb{C}^-, \quad (11)$$

$$\bar{\Delta} \in \Gamma_{\bar{S}}(X; \bar{P}, \bar{Q}), \quad (12)$$

where  $\bar{P} \in \mathbb{R}^{\nu \times \nu}$  and  $\bar{Q} \in \mathbb{R}^{\nu \times \nu}$  are two symmetric positive definite matrices, satisfying a Lyapunov function for  $\bar{A}_*$ , namely:

$$\bar{Q} = -(\bar{P}\bar{A}_* + \bar{A}_*^T\bar{P}), \quad (13)$$

and:

$$\Gamma_{\bar{S}}(X; \bar{P}, \bar{Q}) = \left\{ X \in \mathbb{R}^{\ell \times \nu} \mid \lambda_{Max}(\bar{P}SX + X^T\bar{S}^T\bar{P}) < \lambda_{min}(\bar{Q}) \right\}, \quad (14)$$

(see for example [15], [17]).

Given (9), let us assume that:

$$(\hat{A}, \hat{B}) \text{ is stabilizable, } (\hat{C}, \hat{A}) \text{ is detectable,} \quad (15)$$

in the Hurwitz sense.

Find a linear control law such that the uncertainty signal  $q$  is rejected.

*Remark 2:* In our problem statement we assume the decomposition described by (9), as well as the stabilizability and detectability conditions shown in (15). If the uncertainty signal  $q$  is decoupled via the application of a linear feedback-based controller only depending on the dynamics of  $\hat{x}$  (as well as the reconstruction of  $q$  via an observer-based filter), the resulting closed-loop system can be controlled without taking explicitly into account the unknown dynamic uncertainty.

*Remark 3:* In what follows we shall tackle the solution of our uncertain signal rejection problem in terms of a standard observer-based failure rejection problem. Which is to say, we shall consider systems uncertainty as an actuator fault.

### III. PROBLEM SOLUTION

Let us notice that the state space representation (9) has the standard form used in the fault detection treatment. We then have the following Beard-Jones filter of (9) (cf. [1], [8], [10], [14], [18]):

$$\begin{aligned} d\hat{w}/dt &= (\hat{A} + \hat{K}\hat{C})\hat{w} - \hat{K}y + \hat{B}u, \\ \hat{y} &= \hat{C}\hat{w}, \end{aligned} \quad (16)$$

where  $\hat{K} \in \mathbb{R}^{\hat{n} \times p}$  is an output injection to be computed, such that:

$$\sigma\{\hat{A}_{\hat{K}}\} \subset \mathbb{C}^-, \quad (17)$$

where:

$$\hat{A}_{\hat{K}} = \hat{A} + \hat{K}\hat{C}. \quad (18)$$

We shall now proceed to design an estimator of the failure (i.e., the signal characterizing the failure), via a residual signal generator.

From (9), (16) and (18), we get the following *remainder generator*:

$$\begin{aligned} d\hat{e}/dt &= \hat{A}_{\hat{K}}\hat{e} - \hat{S}q, \\ r &= \hat{C}\hat{e}, \end{aligned} \quad (19)$$

where:  $\hat{e} = \hat{w} - \hat{x}$ ,  $r = \hat{y} - y$ . Let  $G \in \mathbb{R}^{p \times \ell}$  be the steady state gain matrix [4] (recall that  $\hat{A}_{\hat{K}}$  is a Hurwitz matrix):

$$G = F(0) = -\hat{C}\hat{A}_{\hat{K}}^{-1}\hat{S}, \quad (20)$$

where:  $F(s) = \widehat{C}(sI_{\hat{n}} - \widehat{A}_{\widehat{K}})^{-1}\widehat{S}$ . Let us assume that  $F(s)$  is a left invertible transfer matrix [2] of a low pass filter, namely:

$$\ker F(0) = \{0\}, \quad (21)$$

there then exists a  $G^\ell \in \mathbb{R}^{\ell \times p}$  such that:  $G^\ell G = I_\ell$ . Taking into account (20) in (16), we get the following failure estimator:

$$\begin{aligned} d\hat{w}/dt &= (\widehat{A} + \widehat{K}\widehat{C})\hat{w} - \widehat{K}y + \widehat{B}u, \\ \hat{q} &= G^\ell(\widehat{C}\hat{w} - y), \end{aligned} \quad (22)$$

Let us now propose the following state feedback (recall (10)):

$$\hat{u} = u_r + \widehat{F}\hat{x} + \widehat{Q}\hat{q} \quad (23)$$

where  $\widehat{F} \in \mathbb{R}^{\nu \times m}$  is a state feedback to be computed, such that:

$$\sigma\{\widehat{A}_{\widehat{F}}\} \subset \mathbb{C}^-, \quad (24)$$

where:

$$\widehat{A}_{\widehat{F}} = \widehat{A} + \widehat{B}\widehat{F}. \quad (25)$$

From (9), (10), (19), (22), (23) and (25), we get the following closed loop state space representation:

$$\begin{aligned} d\hat{x}/dt &= \widehat{A}_{\widehat{F}}\hat{x} + \widehat{B}u_r + \widehat{S}(q + \hat{q}), \\ d\hat{e}/dt &= \widehat{A}_{\widehat{K}}\hat{e} - \widehat{S}q, \\ \hat{q} &= G^\ell\widehat{C}\hat{e}, \\ y &= \widehat{C}\hat{x}. \end{aligned} \quad (26)$$

The time solution of (26) is:

$$\begin{aligned} y(t) &= y_h(t) + \widehat{C} \int_0^t \exp(\widehat{A}_{\widehat{F}}(t-\tau)) \widehat{B}u_r(\tau) d\tau \\ &\quad + \widehat{C} \int_0^t \exp(\widehat{A}_{\widehat{F}}(t-\tau)) \widehat{S}\tilde{q}(\tau) d\tau, \end{aligned} \quad (27)$$

$$\tilde{q}(t) = q(t) - G^\ell \widehat{C} \int_0^t \exp(\widehat{A}_{\widehat{K}}(t-\tau)) \widehat{S}q(\tau) d\tau, \quad (28)$$

$$y_h(t) = \widehat{C} \left( \exp(\widehat{A}_{\widehat{F}}t) \hat{x}(0) + \widehat{S}G^\ell\widehat{C} \exp(\widehat{A}_{\widehat{F}}t) \hat{e}(0) \right) \quad (29)$$

*Lemma 1:* Under the Hurwitz stability assumptions (11), (12), (17) and (24), the steady state response of  $\tilde{q}$  is:

$$\tilde{q}_{ssr} = \lim_{\substack{dq/dt \rightarrow 0 \\ q(0)=0}} \tilde{q}(t) = 0. \quad (30)$$

Moreover, if the characteristic polynomial of  $\widehat{A}_{\widehat{K}}$  corresponds to a low pass Butterworth filter with corner frequency  $\omega_c$ , namely [5]:

$$\det(\lambda I_{\hat{n}} - \widehat{A}_{\widehat{K}}) = \begin{cases} \prod_{i=1}^{\hat{n}/2} ((\lambda + \sin \theta_i)^2 + \cos^2 \theta_i), & \text{for } \hat{n} \text{ even} \\ (\lambda + 1) \prod_{i=1}^{(\hat{n}-1)/2} ((\lambda + \sin \theta_i)^2 + \cos^2 \theta_i), & \text{for } \hat{n} \text{ odd} \end{cases} \quad (31)$$

where  $\lambda = s/\omega_c$ ,  $\theta_1 = \pi/(2\hat{n})$ ,  $\theta_{j+1} = \theta_j + \Delta\theta$ ,  $\Delta\theta = \pi/\hat{n}$ ,  $j \in \{1, \dots, \hat{n}-1\}$ , there then exist  $k_1, k_2, \alpha_1 \in \mathbb{R}^+$ , such that:

$$\|\tilde{q}(t)\|_2 \leq (k_1/\omega_c) \|dq(t)/dt\|_\infty + k_2 e^{-\alpha_1 t}. \quad (32)$$

*Proof 1:* Let us prove Lemma 1. Integrating by parts (28), we get:

$$\begin{aligned} \tilde{q}(t) &= -G^\ell \widehat{C} \widehat{A}_{\widehat{K}}^{-1} \left( \int_0^t \exp(\widehat{A}_{\widehat{K}}(t-\tau)) \widehat{S} \frac{dq(\tau)}{d\tau} d\tau \right. \\ &\quad \left. + \exp(\widehat{A}_{\widehat{K}}t) \widehat{S}q(0) \right), \end{aligned} \quad (33)$$

Thus, the steady state response of  $\tilde{q}$  satisfies (30).

From (33), we get (see Theorem II.5.10 of [4]):

$$\begin{aligned} \|\tilde{q}(t)\|_2 &\leq k_0 \left\| \widehat{A}_{\widehat{K}}^{-1} \right\|_2 \|dq(t)/dt\|_\infty + k_2 e^{-\alpha_1 t} \\ &\leq k_0 \frac{\left\| \widehat{A}_{\widehat{K}} \right\|_2^{\hat{n}-1}}{\left| \det \widehat{A}_{\widehat{K}} \right|} \left\| \widehat{A}_{\widehat{K}}^{-1} \right\|_2 \|dq(t)/dt\|_\infty + k_2 e^{-\alpha_1 t} \end{aligned} \quad (34)$$

One property of the Butterworth's polynomial (31) is that all its roots are different and equidistant over the left hand side of a circle of radius  $\omega_c$  in the complex plane  $\mathbb{C}$ , which implies that:

$$\left| \det \widehat{A}_{\widehat{K}} \right| = \omega_c^{\hat{n}}, \quad (35)$$

and the canonical Jordan form,  $\widehat{J}_{\widehat{K}}$ , of  $\widehat{A}_{\widehat{K}}$  is a diagonal matrix, so ( $T$  is the basis transformation matrix):

$$\begin{aligned} \left\| \widehat{A}_{\widehat{K}} \right\|_2 &= \left\| T^{-1} \widehat{J}_{\widehat{K}} T \right\|_2 \\ &\leq \|T^{-1}\|_2 \left\| \widehat{J}_{\widehat{K}} \right\|_2 \|T\|_2 \\ &= \|T^{-1}\|_2 \|T\|_2 \omega_c \end{aligned} \quad (36)$$

Thus, from (34), (35) and (36) we get (32).  $\square$

In this way we have proved the following Theorem.

*Theorem 1:* Under the same conditions as in Lemma 1, there exist  $k_3, k_4, \alpha_2 \in \mathbb{R}^+$  such that the closed loop system behaves as:

$$\begin{aligned} \left| y(t) - \widehat{C} \int_0^t \exp(\widehat{A}_{\widehat{F}}(t-\tau)) \widehat{B}u_r(\tau) d\tau \right| &\leq \\ &(k_3/\omega_c) \|dq(t)/dt\|_\infty + k_4 e^{-\alpha_2 t}. \end{aligned} \quad (37)$$

Moreover, the steady state response of  $y$  for high corner frequencies is:

$$y_{ssr} = \lim_{\substack{t \rightarrow \infty \\ \omega_c \rightarrow \infty}} y(t) = \widehat{C} \int_0^\infty \exp(\widehat{A}_{\widehat{F}}(t-\tau)) \widehat{B}u_r(\tau) d\tau. \quad (38)$$

*Remark 4:* Theorem 1 provides a methodology to control the behavior of the system, independently from the presence of dynamic uncertainty, which is considered as fictitious actuators failures. The resulting robust linear control scheme follows.

#### IV. ILLUSTRATIVE EXAMPLE: CONTROL ALTITUDE OF PVTOL AIRCRAFT

In this section we illustrate our robust control methodology, introduced in the previous section, via the regulation of the vertical motion of a PVTOL (Planar Vertical Take-Off and Landing) aircraft in a laboratory setting (see Fig. 1).

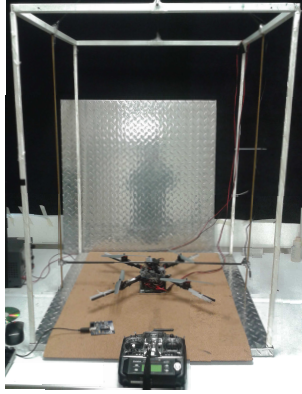


Fig. 1. Laboratory prototype picture.

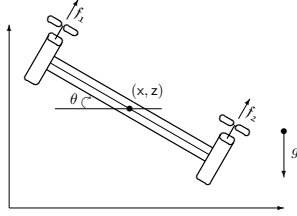


Fig. 2. Schematic PVTOL diagram

#### A. PVTOL aircraft behavioral model

Usually the PVTOL is represented by the following behavioral equations [13] (see Fig. 2):

$$\begin{aligned} m \frac{d^2 z}{dt^2} &= (f_1 + f_2) \cos \theta - mg \\ m \frac{d^2 x}{dt^2} &= (f_1 + f_2) \sin \theta \\ J \frac{d^2 \theta}{dt^2} &= L(f_1 - f_2) \end{aligned} \quad (39)$$

where  $x$ ,  $z$  and  $\theta$  are the horizontal, vertical and angular displacements, respectively.  $m$  is the total mass,  $g$  is the gravity constant,  $J$  is the moment of inertia and  $L$  is the distance from the center of gravity to the thrusters.  $f_1 \geq 0$  and  $f_2 \geq 0$  represent the actuators (thrusters). For our laboratory prototype, we have the parameters given by:  $m = 597 \times 10^{-3}$  [Kg],  $J = 20 \times 10^{-3}$  [Kg m<sup>2</sup>] and  $L = 200 \times 10^{-3}$  m.

As far as the vertical motion is concerned, in theory the PVTOL aircraft simply behaves as a double integrator system, namely ( $\theta = 0$ ):

$$\frac{d^2 z}{dt^2} = \frac{1}{m} u - g \quad (40)$$

where  $u = f_1 + f_2$ . In the laboratory, we have verified that this is not at all true, since there are some proportional and derivative effects which can not be neglected in the takeoff process. Indeed, we have found the following experimental

model for the takeoff process:

$$\frac{d^2 z}{dt^2} + a_1(z) \frac{dz}{dt} + a_2(z) z = \frac{1}{m} u - g \quad (41)$$

where in average, the coefficients  $a_1(z)$  and  $a_2(z)$  behave as (cf. Fig. 3):

$$a_2(z) = \frac{1}{0.3724z + 0.0607}, \quad a_1(z) = \frac{1}{1.3816z + 0.4963} \quad (42)$$

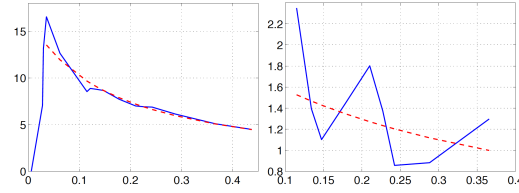


Fig. 3. Behaviors of parameters  $a_2$  and  $a_1$  vs  $\bar{z}$ . Solid blue lines correspond to the experimental values identified by means of the step response. Dotted red lines are curves fitting  $1/a_2(\bar{z})$  and  $1/a_1(\bar{z})$  in a least-square sense. (a) Signal  $a_2$  [1/s<sup>2</sup>] vs  $\bar{z}$  [m]. (b) Signal  $a_1$  [1/s] vs  $\bar{z}$  [m].

*Remark 5:* As we shall show later, signals  $a_1$  and  $a_2$ , depending from the vertical motion, will be described in terms of an actuator fault.

#### B. State Space Representation

A state space representation of (41) is (cf (1) and (2)):

$$\begin{aligned} dx/dt &= (\bar{A} + S \tilde{A}(z)) x + \bar{B}(u/m - g) \\ z &= Cx \end{aligned} \quad (43)$$

where:  $x = [z \quad dz/dt]^T$ , and:<sup>1</sup>

$$\begin{aligned} \bar{A} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \tilde{A}(z) = \begin{bmatrix} -a_2(z) & -a_1(z) \end{bmatrix}, \\ \bar{B} &= S = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned} \quad (44)$$

Let us note that for this laboratory prototype, we have:  $S = \bar{B}$ ,  $\bar{B} = 0$  and  $\mathcal{V}^* = \{0\}$ . Thus:

$$\hat{A} = \bar{A}, \quad \hat{B} = \bar{B}, \quad \hat{S} = \bar{B}, \quad \hat{C} = C, \quad \hat{Q} = 1, \quad (45)$$

and (cf (6)):

$$q = \tilde{A}(z) \hat{x} = \begin{bmatrix} -a_2(z) & -a_1(z) \end{bmatrix} \hat{x}. \quad (46)$$

<sup>1</sup>Let us note that the uncertainties  $a_1(z)$  and  $a_2(z)$  are characterized by matrix  $\tilde{A}(z)$ , which image is contained in  $\text{im } B$ , so  $\text{im } S$  has to be contained in  $\text{im } B$ .

### C. Robust Linear State Feedback

We want to carry the PVTOL from the origin to the altitude  $z_r$ . From (22) and (23), the robust linear control law is:

$$\begin{aligned} d\hat{w}/dt &= (\hat{A} + \hat{K}\hat{C})\hat{w} - \hat{K}\delta z + \hat{B}(\hat{u}/m - g) \\ \hat{q} &= G^\ell(\hat{C}\hat{w} - \delta z) \end{aligned} \quad (47)$$

$$\hat{u} = m(u_r + \hat{F}\hat{x} + \hat{q} + g) \quad (48)$$

where:<sup>2</sup>

$$\begin{aligned} \delta z &= z - z_r, \quad \hat{F} = \begin{bmatrix} \hat{f}_1 & \hat{f}_2 \end{bmatrix}, \quad \hat{K} = \begin{bmatrix} \hat{k}_1 & \hat{k}_2 \end{bmatrix}^T, \\ u_r &= -\hat{f}_1 z_r, \quad G^\ell = \hat{k}_2. \end{aligned} \quad (49)$$

### D. Closed-Loop System

Let us define the following state variables:

$$\hat{\xi} = \begin{bmatrix} \hat{x} \\ \hat{e} \end{bmatrix}, \quad \hat{e} = \hat{w} - \hat{x} + \hat{C}^T z_r. \quad (50)$$

From (43)–(50), the state space representation of the closed loop system is (cf (26) and (46)):

$$\begin{aligned} d\hat{\xi}/dt &= \hat{A}_{cl}\hat{\xi} + \mathbf{f}(\hat{x}) + \hat{B}_{cl}z_r, \\ \begin{bmatrix} z \\ \hat{q} \end{bmatrix} &= \hat{C}_{cl}\hat{\xi}, \end{aligned} \quad (51)$$

where:<sup>3</sup>

$$\begin{aligned} \hat{A}_{cl} &= \begin{bmatrix} \hat{A}_{\hat{F}} & \hat{k}_2 \hat{B} \hat{C} \\ 0 & \hat{A}_{\hat{K}} \end{bmatrix}, \quad \mathbf{f}(\hat{x}) = \begin{bmatrix} \hat{B} \\ -\hat{B} \end{bmatrix} \tilde{A}(z)\hat{x}, \\ \hat{B}_{cl} &= \begin{bmatrix} -\hat{f}_1 \hat{B} \\ 0 \end{bmatrix}, \quad \hat{C}_{cl} = \begin{bmatrix} \hat{C} & 0 \\ 0 & \hat{k}_2 \hat{C} \end{bmatrix}. \end{aligned} \quad (52)$$

### E. Experimental Results

In Figures 4 and 5, we show the experimental results obtained when applying the robust linear control law, (47) and (48), to the laboratory prototype of Figure 2.

The feedback  $\hat{F}$  and the injection  $\hat{K}$  were chosen as (cf (49)):<sup>4</sup>

$$\hat{F} = \begin{bmatrix} -0.1291 & -0.5243 \end{bmatrix}, \quad \hat{K} = \begin{bmatrix} -2.5407 \\ -3.2275 \end{bmatrix}.$$

The reference signal is:

$$z_r = 0.4 \text{ [m]}.$$

The control law is synthesized in a land PC, which communicates to the prototype via a transmitter/receiver XBee Pro s1, operating with a sampling of  $T = 0.01$  [s]. The vertical position is measured by means of the ultrasonic sensor SRF08.

<sup>2</sup>We have translated the state origin (0, 0) to  $(z_r, 0)$ .

<sup>3</sup>Let us note that:  $\hat{C}\hat{C}^T = 1$  and  $\hat{A}\hat{C}^T = 0$ .

<sup>4</sup>The feedback  $\hat{F}$  ( $= \rho^{-1}\hat{B}^T\hat{P}$ ) is obtained from the algebraic Riccati equation:  $\hat{P}\hat{A} + \hat{A}^T\hat{P} - \hat{P}\hat{B}\rho^{-1}\hat{B}^T\hat{P} + Q = 0$ , with:  $Q = I_2$  and  $\rho = 60$ .

The injection  $\hat{K}$  is obtained to satisfy the Butterworth polynomial (31), with:  $\hat{n} = 2$  and  $\omega_c = 1.7965$ ; this value of  $\omega_c$  is set such that the spectrum of  $(\hat{A} + \hat{K}\hat{C})$  is five times faster than the spectrum of  $(\hat{A} + \hat{B}\hat{F})$ .

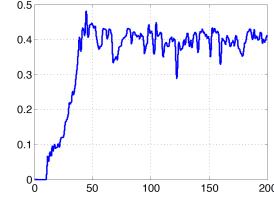


Fig. 4. Vertical position  $z$  [m] vs  $t$  [s].

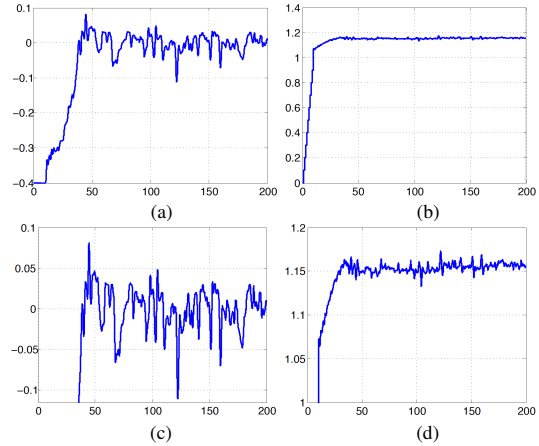


Fig. 5. (a) and (c) Incremental vertical position  $\delta z$  [m] vs  $t$  [s]. (b) and (d) Normalized input control law  $\hat{u}/(mg)$  [-] vs  $t$  [s].

**Remark 6:** As shown in Figure 4, the controller resulting from the application of our robust control methodology successfully regulates the vertical position of the PVTOL aircraft around the requested reference altitude.

**Remark 7:** The vertical position changes between -25% and 20% of the set-point value. The positions changes are due mainly to the altitude sensor used. This sensor could be change of operation range in order to minimizing the drift, but we tried to maintain the highest sample rate.

### V. CONCLUSION

In this paper, we have proposed a robust linear control methodology based on failure detection techniques. We considered continuous-time linear state descriptions, where the matrices  $A$  and  $B$  are affected by linearly structured uncertainty. The systems un-modeled dynamics was characterized in terms of a process fault, which was treated as a fictitious actuator fault. Therefore, the structural uncertainty was rejected with the help of an observer-based failure reconstructor (conceived through the application of concepts coming from the geometric control theoretical framework). The proposed linear robust control scheme was tested via the control altitude of a PVTOL aircraft in a laboratory setting. In this case the structural uncertainty is related to some proportional and derivative effects associated to the takeoff process. Our experimental results confirmed our claims on

the effectivity of our robust control methodology. It must be pointed out that the controller resulting from the application of our control approach is quite easy to implement.

Ease of implementation is then an advantage of our approach. Moreover, taking into account the systems unmodeled dynamics in terms of structural uncertainty affecting the state description of the system, led us to tackle the control problem in a quite simple way. In our illustrative example the control problem is established as a stabilization problem (the requested vertical position of the PVTOL aircraft is fixed as the state origin). The uncertainty modeling approach that we followed here avoided us to consider standard  $H_\infty$  control and identification techniques based on input-output uncertainty modeling, mainly developed in input/output terms ([16], [11]).

The proposed methodology is limited to linear systems with structured uncertainties, characterized by (2) and satisfying (3). The analysis followed in this paper is more or less standard in the geometric approach, so it should not be so complicated to extend these results to the nonlinear case. We have used a cheapest laboratory prototype of a PVTOL to test our proposition. Actually we are testing our results in a quadrotor with a better instrumentation set. We will do experiments on the environment, in the presence of disturbances like the environmental wind.

#### REFERENCES

- [1] BEARD, R.V. (1971). Failure accommodation in linear systems through self-reorganization. (*PhD thesis, Massachusetts Institute of Technology, 1971.*)
- [2] G. BASILE, G. AND MARRO, G. (1992). Controlled and Conditioned Invariants in Linear System Theory. *New York: Prentice-Hall.*
- [3] BOUABDALLAH, S., AND SIEGWART, R. (2007, October). Full control of a quadrotor. In Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on (pp. 153-158). IEEE.
- [4] DESOER, C.A. AND VIDYASAGAR, M. (1975). *Feedback Systems: Input-Output Properties.* Academic Press.
- [5] DANIELLS R.(1974). *Approximation Methods for Electronic Filter Design.* McGraw-Hill Book Company, Inc.
- [6] EFE, M. Ö. (2007, June). Robust low altitude behavior control of a quadrotor rotorcraft through sliding modes. In Control & Automation, 2007. MED'07. Mediterranean Conference on (pp. 1-6). IEEE.
- [7] HOFFMANN, G. M., HUANG, H., WASLANDER, S. L., AND TOMLIN, C. J. (2011). Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control engineering practice*, 19(9), 1023-1036.
- [8] ISERMANN, R. (1984). Process fault detection based on modeling and estimation methods-A survey, *Automatica*, 20, 387-404.
- [9] LEE, K. U., YUN, Y. H., CHANG, W., PARK, J. B., AND CHOI, Y. H. (2011, October). Modeling and altitude control of quad-rotor UAV. In Control, Automation and Systems (ICCAS), 2011 11th International Conference on (pp. 1897-1902). IEEE.
- [10] MASSOUMNIA, M.-A. (1986). A geometric approach to the synthesis of failure detection filters. *IEEE Trans. Automatic Control*, 31(9), 1986, 839-846.
- [11] LI, Z., MAZARS, E., ZHANG, Z., JAIMOUKHA, I. M. (2011). State-space solution to the  $H_\infty/H^\infty$  fault-detection problem. *Int. J. Robust. Nonlinear Control* 2012; 22:282299.
- [12] PARK, S., WON, D. H., KANG, M. S., KIM, T. J., LEE, H. G., AND KWON, S. J. (2005, August). Ric (robust internal-loop compensator) based flight control of a quad-rotor type uav. In Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 3542-3547). IEEE.
- [13] POLDERMAN, J.W., WILLEMS, J.C. (1998). Introduction to Mathematical Systems Theory: A Behavioral Approach. *New York: Springer-Verlag.*
- [14] SABERI, A., STOORVOGEL, A.A., SANNUTI, P. & NIEMANN, H.H. (2000). Fundamental problems in fault detection and identification. *International Journal of Robust and Nonlinear Control*, 10, 1209-1236.
- [15] STEWART, G. W. (1973). Introduction to Matrix Computations. *New York: Academic Press.*
- [16] TRENTELMAN, H. L., STOORVOGEL, A. A., HAUTUS, M. (2001). Control theory for linear systems. New York: Springer.
- [17] VIDYASAGAR, M. (2002). Nonlinear Systems Analysis (Classics in Applied Mathematics) (No. 42) 2nd Edition. *Philadelphia, PA: Society for Industrial and Applied Mathematics.*
- [18] WILLISKY, A.S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, 12, 601-611.
- [19] WONHAM, W.M. (1985). Linear Multivariable Control: A Geometric Approach. *New York: Springer-Verlag, 3rd ed.*

# Bibliografía

- [1] Timothy H. Cox, Christopher J. Nagy, Mark A. Skoog, Ivan A. Somers, and Ryan Warner. Civil UAV capability assessment. *NASA, Tech. Rep., draft Version*, 2004.
- [2] Hazim Shakhatreh, Ahmad H. Sawalmeh, Ala Al-Fuqaha, Zuocho Dou, Eyad Al-maita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634, 2019.
- [3] Zak Sarris and S. Atlas. Survey of UAV applications in civil markets. In *IEEE Mediterranean Conference on Control and Automation*, volume 11, 2001.
- [4] M. I. Ltd. Maverick industrial UAV inspection. <http://www.maverickinspection.com/>, 1994. Accedido 30-07-2018.
- [5] Dá-Jiāng Innovations Science and Technology Co. Ltd. Agrasmg-1. <https://www.dji.com/es/mg-1>, 2006. Accedido 30-07-2018.
- [6] Wei Dong, Guo-Ying Gu, Xiangyang Zhu, and Han Ding. Modeling and control of a quadrotor UAV with aerodynamic concepts. In *Proceedings of World Academy of Science, Engineering and Technology*, number 77, page 437. World Academy of Science, Engineering and Technology (WASET), 2013.
- [7] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6461, 2007.

- [8] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The grasp multiple micro-uav testbed. *IEEE Robot. Automat. Mag.*, 17(3):56–65, 2010.
- [9] İ. Can Dikmen, Aydemir Arisoy, and Hakan Temeltas. Attitude control of a quadrotor. In *2009 4th International Conference on Recent Advances in Space Technologies*, pages 722–727. IEEE, 2009.
- [10] Samir Bouabdallah, Andre Noth, and Roland Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2451–2456. IEEE, 2004.
- [11] Jun Li and Yuntang Li. Dynamic analysis and PID control for a quadrotor. In *2011 IEEE International Conference on Mechatronics and Automation*, pages 573–578. IEEE, 2011.
- [12] Tarek Madani and Abdelaziz Benallegue. Backstepping control for a quadrotor helicopter. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3255–3260. IEEE, 2006.
- [13] Xing Huo, Mingyi Huo, and Hamid Reza Karimi. Attitude stabilization control of a quadrotor UAV by using backstepping approach. *Mathematical Problems in Engineering*, 2014:9, 2014.
- [14] Guilherme V. Raffo, Manuel G. Ortega, and Francisco R. Rubio. An integral predictive/nonlinear  $\mathcal{H}_\infty$  control structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, 2010.
- [15] Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, 25(6):45–55, 2005.
- [16] J. Escareno, S. Salazar-Cruz, and R. Lozano. Embedded control of a four-rotor UAV. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

- [17] Daewon Lee, H Jin Kim, and Shankar Sastry. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of control, Automation and systems*, 7(3):419–428, 2009.
- [18] Qing-Li Zhou, Youmin Zhang, Camille-Alain Rabbath, and Didier Theilliol. Design of feedback linearization control and reconfigurable control allocation with application to a quadrotor UAV. In *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 371–376. IEEE, 2010.
- [19] Hao Liu, Yongqiang Bai, Geng Lu, Zongying Shi, and Yisheng Zhong. Robust tracking control of a quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, 75(3-4):595–608, 2014.
- [20] David Cabecinhas, Rita Cunha, and Carlos Silvestre. A nonlinear quadrotor trajectory tracking controller with disturbance rejection. *Control Engineering Practice*, 26:1–10, 2014.
- [21] Yong Zhang, Zengqiang Chen, Xinghui Zhang, Qinglin Sun, and Mingwei Sun. A novel control scheme for quadrotor UAV based upon active disturbance rejection control. *Aerospace Science and Technology*, 79:601–609, 2018.
- [22] Alberto Castillo, Ricardo Sanz, Pedro Garcia, Wei Qiu, Hongda Wang, and Chao Xu. Disturbance observer-based quadrotor attitude tracking control for aggressive maneuvers. *Control Engineering Practice*, 82:14–23, 2019.
- [23] Luis Rodolfo García Carrillo, Alejandro Enrique Dzul López, Rogelio Lozano, and Claude Pégard. *Quad rotorcraft control: vision-based hovering and navigation*. Springer Science & Business Media, 2012.
- [24] Michael V. Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Elsevier, 3rd edition, 2013.
- [25] Robert F. Stengel. *Flight dynamics*. Princeton University Press, 2015.



- 
- [26] W. M. Wonham. *Linear Multivariable Control: A Geometric Approach*. New York: Springer-Verlag, 3rd edition, 1985.
- [27] Pavol Brunovsky. A classification of linear controllable systems. *Kybernetika*, 6(3):173–188, 1970.
- [28] Jan C. Willems. Input-output and state-space representations of finite-dimensional linear time-invariant systems. *Linear Algebra and its Applications*, 50:581–608, 1983.
- [29] Jan C. Willems and Jan W. Polderman. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. New York: Springer-Verlag, 1998.
- [30] M. Bonilla E., L.A. Blas, J.C. Martínez, S. Salazar, and M. Malabre. A robust linear control methodology based on fictitious failure rejection. In *2016 European Control Conference (ECC)*, pages 2596–2601. IEEE, 2016.
- [31] Richard Vernon Beard. *Failure accomodation in linear systems through self-reorganization*. PhD thesis, Massachusetts Institute of Technology, 1971.
- [32] Rolf Isermann. Process fault detection based on modeling and estimation methods - A survey. *Automatica*, 20(4):387 – 404, 1984.
- [33] M.-A. Massoumnia. A geometric approach to the synthesis of failure detection filters. *IEEE Transactions on Automatic Control*, 31(9):839–846, 1986.
- [34] Ali Saberi, Anton A Stoorvogel, Peddapullaiah Sannuti, and Henrik Niemann. Fundamental problems in fault detection and identification. *International Journal of Robust and Nonlinear Control*, 10(14):1209–1236, 2000.
- [35] Alan S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601 – 611, 1976.
- [36] M. Bonilla, L.A. Blas, V. Azhmyakov, M. Malabre, and S. Salazar. Robust structural feedback linearization based on the nonlinearities rejection. *Journal of the Franklin Institute*, 2019.
-

- [37] Mathukumalli Vidyasagar. *Nonlinear Systems Analysis*, volume 42. Siam, 2002.
- [38] Henri P. Gavin, Rodrigo Morales, and Kathryn Reilly. Drift-free integrators. *Review of Scientific Instruments*, 69(5):2171–2175, 1998.
- [39] Paul Horowitz and Winfield Hill. *The art of electronics*. Cambridge Univ. Press, 2nd edition, 1989.
- [40] David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., New York, 1969.
- [41] David Morin. *Introduction to classical mechanics: with problems and solutions*. Cambridge University Press, 2008.