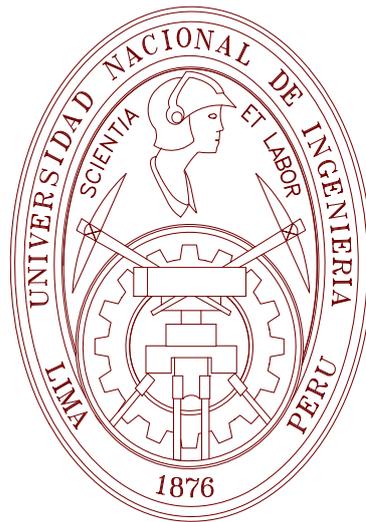


**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERIA MECÁNICA**



**DISEÑO DE UN SIMULADOR DE VUELO Y CONTROL  
DE POSICIÓN PARA UN MINI VEHÍCULO AÉREO**

**TESIS**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:  
INGENIERO MECATRÓNICO**

**NOMBRES Y APELLIDOS:**

**RUDOLPH HIMBLER MOLERO FERNANDEZ**

**PROMOCIÓN 2008-I**

**LIMA-PERÚ**

**2010**

*Dedico la presente tesis en primer lugar a Dios, por haberme permitido continuar en todo momento firme en mis objetivos. En segundo lugar agradezco a mi familia, por haberme brindado el apoyo incondicional durante toda mi vida.*

*En tercer lugar agradezco a todas mis amistades, por haberme brindado su preciada ayuda siempre. Quiero también agradecer muy especialmente al Grupo GISCIA, por concederme su amistad, a mis dos asesores Mario Borja y Victor Sotelo, por sus sabios e incisivos consejos, a mis asesores de la Universidad Carnegie Mellon Sebastian Scherer, Lyle Chamberlain y Sajiv Singh, por haberme otorgado el placer de investigar con ellos, a mis amigos Martín, Nilton, Drago, Edgar, Javier, José Luis, Anita y al Ing. Hugo Fernández, por su eterna confianza depositada en mí.*

# Índice general

<b>Prólogo</b>	<b>1</b>
<b>1. Introducción</b>	<b>4</b>
1.1. Motivación . . . . .	4
1.1.1. Aspectos importantes . . . . .	5
1.2. Personajes Relacionados . . . . .	7
1.2.1. Leonhard Paul Euler . . . . .	7
1.2.2. Aleksandr Mikhailovich Lyapunov . . . . .	9
1.2.3. Igor Sikorsky . . . . .	10
1.2.4. Rudolf Emil Kalman . . . . .	13
1.2.5. Gareth Padfield . . . . .	14
1.2.6. Bernard Mettler . . . . .	15
1.3. Estado del Arte . . . . .	16
1.3.1. Definición de vehículos aéreos no tripulados (UAV) . . . . .	16
1.3.2. Antecedentes históricos . . . . .	17
1.3.3. Trabajos relacionados . . . . .	34
1.3.4. Aplicaciones de los vehículos no tripulados . . . . .	42
1.4. Teoría general del helicóptero . . . . .	47
1.4.1. Transmisión en el rotor principal . . . . .	49
1.4.2. Flapping o aleteo . . . . .	51
1.4.3. Movimiento Pitch . . . . .	51

1.5. Objetivos . . . . .	53
1.6. Alcances . . . . .	54
<b>2. Marco Teórico</b>	<b>55</b>
2.1. Sistema de coordenadas . . . . .	55
2.1.1. Angulos de Euler . . . . .	55
2.1.2. Rotaciones de Euler . . . . .	58
2.2. Parámetros del mini helicóptero Caliber 5 . . . . .	60
2.3. Ecuaciones de cuerpo rígido . . . . .	63
2.3.1. Ecuaciones de Newton-Euler . . . . .	63
2.4. Modelo aerodinámico del mini helicóptero . . . . .	65
2.4.1. Ecuaciones de movimiento . . . . .	67
2.4.2. Influjo ( <i>Inflow</i> ) del rotor . . . . .	69
2.4.3. Fuerzas y momentos en el rotor principal . . . . .	70
2.4.4. Modelo del Motor, gobernador ( <i>governor</i> ) y velocidad del rotor . . . . .	80
2.4.5. Fuerzas en el fuselaje . . . . .	83
2.4.6. Rotor de la cola . . . . .	84
2.4.7. Momentos y fuerzas en el fin vertical . . . . .	87
2.4.8. Fuerzas y momentos en el estabilizador horizontal . . . . .	89
2.4.9. Ecuaciones integradas del helicóptero . . . . .	89
2.5. Emparejamiento de barrido estándar . . . . .	91
2.5.1. Algoritmo iterative closest point (ICP) . . . . .	92
2.6. Conceptos básico de probabilidad . . . . .	95
2.7. Predictor de Kalman . . . . .	96
2.7.1. El algoritmo del filtro de Kalman . . . . .	98
2.7.2. El predictor de Kalman . . . . .	99

<b>3. Diseño del Software de Simulación de Vuelo</b>	<b>100</b>
3.1. Diseño de entorno en realidad virtual . . . . .	100
3.1.1. Ajuste de entorno 3D . . . . .	101
3.1.2. Función <i>initRendering()</i> . . . . .	104
3.1.3. Función <i>drawScene()</i> . . . . .	106
3.1.4. Función <i>handleKeypress()</i> . . . . .	110
3.1.5. Función <i>specialFunc()</i> . . . . .	111
3.1.6. Función <i>handleResize()</i> . . . . .	112
3.1.7. Función <i>control()</i> . . . . .	112
3.1.8. Solución Numérica de Ecuaciones Diferenciales . . . . .	114
3.1.9. Resultados del Simulador . . . . .	117
<b>4. Diseño del Sensor de Posición y Orientación Basado en Empare-</b>	
<b>    jamiento de Barrido Láser</b>	<b>119</b>
4.1. Búsqueda en la vecindad . . . . .	120
4.2. Predicción de movimiento . . . . .	122
4.3. Técnica de la pose más probable . . . . .	123
4.4. Rechazo de barrido basado en la actitud . . . . .	125
4.5. Optimizaciones . . . . .	126
<b>5. Control de Posición de un Mini Vehículo Aéreo</b>	<b>129</b>
5.1. Arquitectura de Software . . . . .	129
5.1.1. Implementación del algoritmo de estimación de pose . . . . .	129
5.1.2. Controlador de posición . . . . .	130
5.2. Hardware . . . . .	131
5.3. Experimentos de control . . . . .	133
5.3.1. Configuración . . . . .	133
5.3.2. Resultados . . . . .	134

<b>Conclusiones</b>	<b>138</b>
<b>Recomendaciones</b>	<b>140</b>
<b>Bibliografía</b>	<b>142</b>
<b>A. Especificaciones técnicas del pico-computador VIA EPIA PX10000G</b>	
<b>Pico-ITX</b>	<b>144</b>

# Índice de figuras

1.1. Labores de mantenimiento e inspección de tendidos de alta tensión desde un vehículo aéreo. Para mayor información visitar <a href="http://www.fkec.com/">http://www.fkec.com/</a> . . . . .	7
1.2. Leonhard Paul Euler. . . . .	8
1.3. Aleksandr Mikhailovich Lyapunov. . . . .	9
1.4. Igor Sikorsky (1889-1972). A su llegada a los Estados Unidos procedente de Rusia en 1919, más tarde dijera “ <i>Encontré lo que había esperado, un país dinámico, enérgico y progresista.</i> ” . . . . .	11
1.5. Hidroavión S-26 de Sikorsky. . . . .	12
1.6. Rudolf Emil Kalman. . . . .	13
1.7. Gareth Padfield. . . . .	14
1.8. Bernard Mettler. . . . .	16
1.9. Ilustración de un artista de la paloma voladora, el primer UAV documentado de la historia. Se ha reportado que voló por cerca de 200 m. . . . .	18
1.10. Representación mecánica de un objeto volador. El eternamente juguete entraña principios en los que se bases el funcionamiento de todos los helicóptero modernos. . . . .	19
1.11. Ave voladora con un mecanismo en el estómago, atribuida a un ingeniero renacentista. . . . .	19

- 1.12. Propulsor de aire de Leonardo da Vinci. Un plano continuo en espiral impartía fuerza ascensional al aparato; la energía procedía del un sencillo mecanismo de relojería. Si el mecanismo tuvo éxito, como dejan traslucir las notas de Leonardo, fue el primer mecanismo aéreo que voló con una fuente de energía incorporada en sí mismo. . . . . 20
- 1.13. Carruaje aéreo de George Cayley. . . . . 21
- 1.14. Una irónica visión del siglo XX nos muestra a unos respetables ciudadanos paseando plácidamente por el aire portando rotores sujetos a la espalda con correas. . . . . 22
- 1.15. Gabriel de La Landelle ideó en 1863 esta “nave aérea de vapor” que usaba ocho rotores elevadores y dos mástiles, cada uno de ellos coronado por un paracaídas de paraguas. También tenía dos grandes alas, un hélice a popa y una góndola colgante para los pasajeros. . . . . 22
- 1.16. Helicóptero de Ponton d’Amécourt. . . . . 23
- 1.17. Helicóptero de Paul Cornu. . . . . 24
- 1.18. Frente a la casa de su padre en Kiev, en 1910, el joven Sikorsky se apresta a hacer algunos ajustes a su segundo helicóptero. Reconstruido a partir de un modelo de 1909 el cual fuera inviable por sus muchas vibraciones, el nuevo modelo pesaba menos, y en palabras del propio Sikorsky, se parecía a “*una enorme mariposa*”. Podía elevar su propio peso (181 Kg) pero nada más: ni combustible para sostener el vuelo ni piloto para gobernarlo. . . . . 25
- 1.19. El helicóptero de Stephan Petroczy y Theodore Von Karman consistente de dos propulsores sustentadores superpuestos. . . . . 25
- 1.20. El helicóptero de Bothezat con seis rotores de palas (1922) fue respaldado por la armada estadounidense. . . . . 26

1.21. El autogiro de Juan de la Cierva fue un dispositivo híbrido con alas y cola, pero también con un rotor vertical sobre el fuselaje. . . . .	26
1.22. El helicóptero Fa-61 (1936) fue una máquina de dos rotores de lado a lado. Este es el primer helicóptero que demostró un completo control de vuelo y exitosas auto-rotaciones. . . . .	27
1.23. Cerniéndose tan abajo y tan inmóvil que un ayudante podía colocar un paquete en un canastilla acoplado a la proa de la aeronave, Sikorsky demuestra la maniobrabilidad de su VS-300. Los pontones acoplados a los patines del helicóptero hacían que pudiera utilizarse tanto desde tierra como desde agua. . . . .	27
1.24. Al colocar el motor en la proa del S-55, un helicóptero que se usó mucho en Korea, Sikorsky consiguió espacio para una cabina con capacidad para 10 personas. El motor se conectaba al rotor por medio de un árbol de transmisión inclinado que pasaba entre los asientos de una carlinga. El aparato que se reproduce aquí luce los colores de la Marina Real canadiense que utilizó el S-55 para misiones de rescate. . . . .	28
1.25. Firebee fabricado por NORTHROP GRUMMAN. . . . .	29
1.26. Compass COPE actualmente retirado al museo de la USAF (United States Air Force). . . . .	29
1.27. Dragon Drone usado por la U.S. Army (Armada de los Estados Unidos) y por la Marin Corps (Infantería de Marina). . . . .	30
1.28. Eagle Eye de Bell Helicopter. . . . .	31
1.29. Antenas de seguimiento del Eagle Eye. . . . .	31
1.30. CL-227 Sentinel de la Fuerza Aérea de Canadá. . . . .	32
1.31. Se muestra el Phampton Sentinel cuando es lanzado por un soldado. . . . .	32
1.32. Global Hawk de NORTHROP GRUMMAN. . . . .	33
1.33. Predator de la USAF. . . . .	33

1.34. RMAX de Yamaha el cual es muy usado en proyectos de UAVs. . . . .	34
1.35. Equipo de la universidad Carnegie Mellon con el RMAX de Yamaha. . . . .	37
1.36. Configuración del sistema STARMAC de la Universidad de Stanford. . . . .	41
1.37. El quadrotor equipado con un sistema de navegación durante una misión (arriba). La posición estimada on-line durante el vuelo (abajo). . . . .	42
1.38. Helicóptero quadrotor realizando SLAM. . . . .	42
1.39. Mini helicóptero realizando inspección en una presa. . . . .	44
1.40. Mini helicóptero realizando tarea de búsqueda. . . . .	44
1.41. Mini helicóptero realizando patrullaje (izquierda). Mini helicóptero realizando persecución policíaca (derecha). . . . .	45
1.42. Mini helicóptero realizando mapeo aéreo. . . . .	46
1.43. Mini helicóptero realizando tomas aéreas para una película. . . . .	47
1.44. Mini helicóptero realizando tareas de fumigación. . . . .	47
1.45. Componentes de para el modelado de un helicóptero . . . . .	48
1.46. Ilustración del roll, pitch y yaw en el fuselaje del helicóptero. . . . .	49
1.47. Se muestra la transmisión mecánica del swashplate con la palas del rotor principal, el cual por este medio controla el pitch del las palas. . . . .	50
1.48. Se muestra la barra estabilizadora, la cual proporciona estabilidad al vuelo de los mini helicópteros. . . . .	51
1.49. Los puntos extremos de las palas forman un plano conocido como TPP. El HP es igual al TPP cuando no hay flapping. . . . .	52
1.50. Vista superior del swash plate y la conexión a la palas. . . . .	52
1.51. Vista de perfil de swash plate, sus conexiones y transmisiones. . . . .	53
2.1. Angulos de Euler vistos de acuerdo a las rotaciones en los 3 ejes . . . . .	56
2.2. Rotaciones de Euler de acuerdo a la secuencia estándar. . . . .	59
2.3. Momentos y Fuerzas actuando en el Helicóptero . . . . .	68

2.4. Estados del flujo en el rotor en movimiento axial: (a)suspensión (hover); (b) ascenso (climb); (c) descenso (descent) . . . . .	69
2.5. Las flechas curvadas indican el flujo de aire alrededor del disco del rotor. El helicóptero mostrado es el RAH-66 Comanche . . . . .	72
2.6. Representación del Tip-Path-Plane en el rotor. El ángulo $\beta$ está representado por la serie de Fourier, con término no periódico $\beta_0$ el cual describe la conicidad y los primeros armónicos $\beta_{1s}$ (o $\beta_{1c}$ ) describen la inclinación lateral (longitudinal) del angulo del TPP. . . . .	75
2.7. Momentos del rotor actuando sobre el fuselaje . . . . .	77
2.8. Vortices wake en modo avance . . . . .	86
2.9. Modelo integrado de simulación del helicóptero. . . . .	90
2.10. Funcionamiento general del sensor ladar. . . . .	91
2.11. Visión general del estándar ICP. . . . .	93
2.12. Consideración de correspondencia del punto más cercano. . . . .	93
2.13. Diagrama de flujo del algoritmo de emparejamiento de barrido. . . . .	94
3.1. Imagen base del piso del HeliSi3D (tamaño=256x256 Pixeles) . . . . .	106
3.2. Piso del entorno de animación . . . . .	107
3.3. Simulador HeliSi3D . . . . .	116
3.4. Comportamiento gráfico del simulador HeliSi3D para un ángulo de ataque en el rotor principal de $\Theta = 4.0^\circ$ , un ángulo de ataque en el rotor de la cola de $\Theta_{tr} = -10.0^\circ$ , una inclinación del swashplate de 50% en la dirección lateral, y una posición inicial ubicada en el origen de coordenadas. El subíndice $f$ indica que el valor corresponde a la cantidad de la coordenada final de la gráfica. . . . .	118
4.1. Algoritmo ICP y modificaciones. MPP (most probable pose technique) o técnica de la pose más probable. . . . .	120

4.2. Determinación de la distancia más corta de un punto en la referencia  $M_i$  a la escena  $S_i$ . . . . . 121

4.3. Esbozo general de la búsqueda en la vecindad. . . . . 122

4.4. Filtro de Kalman es configurado para estimar  $x$ ,  $y$  y orientación. Esto permite tener dos cosas: un tanteo inicial para la matriz de transformación y un error convergencia inicial para el algoritmo ICP. 123

4.5. Las estimaciones de orientación  $\theta$  pueden ser hipotéticamente acomodadas en la campana de Gauss; a fin de mostrar la exclusión de estimaciones alejadas de las estimaciones más probables. . . . . 125

4.6. En la figura de arriba, el quadrotor aparece equipado y sobrevolando en modo manual. En la figura de abajo, vemos que el barrido de referencia que corresponde a ese instante de tiempo es rechazado debido a que su ángulo roll es menor a  $-5^\circ$ . . . . . 127

5.1. Diagrama de flujo del algoritmo de emparejamiento de barrido y su función es representada por **SM(Modelo,Escena)**. La variable **contador** es usada para guardar el orden en los juegos de datos entrantes. . . . . 131

5.2. Diagrama de bloques del controlador de posición. . . . . 132

5.3. Helicóptero quadrotor equipado con un pico-computador y un sensor láser. . . . . 133

5.4. Configuración inicial del experimento de control: quadrotor encima de la referencia visual. . . . . 135

5.5. Control Hover con perturbaciones manuales. . . . . 136

5.6. Señales y variables de control. . . . . 137

# Índice de tablas

2.1. Parámetros del mini helicóptero Caliber 5 . . . . .	61
2.2. Especificaciones del motor de combustion MAX-50SX-HRING HYPER . . . . .	81
2.3. Algoritmo del filtro de Kalman para estados de transición y mediciones Gaussianas lineales . . . . .	98
2.4. Algoritmo del predictor de Kalman . . . . .	99

# Prólogo

Actualmente el Perú se encuentra en una creciente necesidad de utilizar sistemas aéreos autónomos o remotamente controlados con objetivos civiles o militares. Estos sistemas podrían operar en misiones de inspección, monitoreo, escaneamiento o vigilancia. Por eso, es de vital importancia que se implementen estos sistemas ya que el País necesita, por ejemplo, inspeccionar rápidamente zonas de desastres, detectar focos del crimen organizado, u obtener rápidamente mapas tridimensionales de determinadas zonas con lo cual se contribuye enormemente al desarrollo del País.

Las teorías más importantes necesarias en la presente tesis son la dinámica de vuelo, la ingeniería de control y la robótica probabilística. El dominio de estas teorías es de suma importancia para modelar y controlar los sistemas de vuelo como el helicóptero.

En la dinámica del vuelo, encontramos los ángulos y rotaciones de Euler, y la dinámica de Newton-Euler. Es vital plantear una adecuada relación entre el sistema de referencia inercial y el sistema de referencia no inercial, pues de ello van a depender todas la ecuaciones que en adelante van a influir en el modelado y simulación del mini-helicóptero. Una vez que ya lo hemos referenciado apropiadamente, la dinámica de Newton-Euler nos da las ecuaciones básicas del sistema. Dichas ecuaciones advierten la existencia de 3 fuerzas (en los ejes xyz referenciales) y 3 momentos (en los ejes xyz referenciales), los cuales van a ser tratados en el análisis aerodinámico y mecánico de todos los componentes del

helicóptero.

Cabe mencionar que muchas de las características y aspectos que se van a discutir son del tipo experimentales. Debido a la complejidad de las fuerzas presentes en el rotor principal, no es viable realizar un análisis minucioso de todas la fuerzas presentes. Por lo cual, muchos de los parámetros y el modelo mismo van a estar supeditados a ciertos rangos de trabajo o condiciones de vuelo.

Debido también a un creciente empleo de métodos estadísticos en la robótica, esta tesis ha optado por utilizar muchos de los métodos disponibles en esta área. Esto que fue posible gracias a la invitación de continuar con dicha investigación en el Instituto de Robótica de la Universidad Carnegie Mellon, USA. Los detalles y fundamentos se verán en los capítulos subsiguientes.

Esta tesis, a modo también de manual, está dirigida como una guía de futuros trabajos acerca del vuelo del helicóptero. Permitiendo así un entendimiento rápido de las principales características que involucran a los vehículos aéreos. La evidencia de procedimientos ya probados y verificados va a ser de vital importancia para ofrecer un compendio base necesario para los futuros investigadores que deseen emplear los métodos aquí expuestos.

El presente trabajo está dividido en cinco capítulos que son detallados a continuación:

**Capítulo 1:** contiene la motivación de la tesis, los personajes históricos que contribuyeron con sus estudios y herramientas matemáticas, el estado de arte del desarrollo de los vehículos autónomos, y finalmente un alcance básico del funcionamiento de los helicópteros.

**Capítulo 2:** contiene el fundamento teórico de la dinámica y aerodinámica de vuelo de un helicóptero de escala pequeña. También, contiene el trasfondo matemático de la estimación de pose (posición y orientación en el plano) con un sensor láser de escaneo de contornos aledaños. Finalmente, muestra algunos métodos probabilísticos empleados en la investigación.

**Capítulo 3:** contiene el diseño del simulador tridimensional en una plataforma libre de animación tridimensional llamada OpenGL. Por medio del detalle de cada función de programación utilizada y de la forma cómo las ecuaciones de estado del sistema son empleadas, se llega a culminar el capítulo con una exhibición del simulador de vuelo HeliSi3D.

**Capítulo 4:** este capítulo se ocupa de mostrar las mejoras del método básico del algoritmo de emparejamiento de barrido expuesto en el capítulo 2. Siendo estas mejoras: rechazo de barrido, búsqueda en la vecindad, predicción de movimiento y la técnica de la pose más probable. El capítulo acaba mostrando la optimizaciones del algoritmo de estimación de pose luego de añadir las modificaciones.

**Capítulo 5:** trata del control de posición del helicóptero quadrotor usando el algoritmo de estimación de pose detallado en los capítulos 2 y 4. También muestra el proceso de implementación y las pruebas finales de laboratorio, validando así las modificaciones al algoritmo básico de emparejamiento de barrido.

# Capítulo 1

## Introducción

### 1.1. Motivación

Los vehículos no tripulados han desempeñado un rol crítico en el desarrollo de actividades de rescate, vigilancia y salvamento. Sin embargo, en un ambiente urbano como en los que tenemos en nuestro país, es necesaria la presencia de vehículos ágiles capaces de movilizarse con un cierto grado de autonomía [5, p. 13].

Los helicópteros en miniatura como los CALIBER 5 o los quadrotores tienen una alta concentración de rigidez y una elevada relación empuje-peso, lo cual hace que estos helicópteros tengan una agilidad natural. Algunos mini-helicópteros pueden presentar impulsos de 2 a 3 veces el peso de los mismos [5, p. 13]. La alta rigidez del rotor principal permite que estos helicópteros tengan una respuesta dinámica bastante rápida comparado con los helicóptero de escala real.

Con la habilidad de despegar y aterrizar de manera vertical, estos vehículos de rotor pueden desempeñar múltiples tareas autónomas. Por ejemplo, pueden alertar de accidentes o inspeccionar zonas de difícil acceso [13, p. 3]. Entre otras aplicaciones tenemos la elaboración de películas (permitiendo vistas estables y en movimiento), la inspección de puentes, líneas de transmisión, construcciones y

represas, y el modelado tridimensional de terrenos donde un helicóptero pequeño dada la cercanía a la que este se puede aproximar de su objetivo, puede conseguir mayores detalles del entorno que si fuera hecho con una aeronave. También se ha propuesto usar estos mini helicópteros para futuras exploraciones en Marte [13, p. 3]. Dichas exploraciones serían hechas con mucha maniobrabilidad dado que los vehículos terrestres han presentado ciertas complicaciones en terrenos muy agrestes o toscos.

De manera concreta los helicópteros son difíciles de volar, pues son de carácter inestable. Es decir, requieren de una atención constante de parte del piloto. Cuatro son las entradas de control principales en un helicóptero: el cíclico longitudinal y lateral; colectivo para su desplazamiento vertical; y el pedal para el movimiento en la dirección *Yaw* (guiñada). El helicóptero es un ejemplo claro de un sistema multivariable pues el piloto en todo momento debe de compensar las componentes de segundo orden que se generan.

Varios sistemas de control se han aplicado a helicópteros de escala real, los cuales presentaban autonomía limitada en ciertas de regiones de trabajo. Estos sistemas se hicieron utilizando la teoría de control clásico. Sin embargo, en nuestro caso el modelo que vamos a plantear va a contener un sistema de control capaz de aprovechar en mayor medida la agilidad del mini-helicóptero [13, p. 15]. Para esto se planteará un controlador adecuado que será validado con las simulaciones y animaciones, y que posteriormente será corroborado en la implementación.

### **1.1.1. Aspectos importantes**

En la actualidad hay muchas situaciones riesgosas donde no es viable el empleo de seres humanos. Muchas inspecciones civiles requieren que se instalen grandes andamios o que se cuelguen hombres. En otras situaciones, el hecho de acceder por tierra al lugar en cuestión hace también que una tarea simple se torne costosa.

Por otro lado, la supervisión o la vigilancia de amplias extensiones requiere el traslado de personal y equipo apropiado lo cual hace que el proceso de inspección terrestre se torne infructuoso e ineficiente [6, p. 1].

Debido a las dificultades mencionadas, es que surge la inspección aérea. En la actualidad la mayoría de estas tareas han sido hechas por helicóptero tripulados (ver figura 1.1), ya que estos cuentan con la capacidad de realizar vuelos estáticos y la posibilidad de movilizarse a bajas velocidades [6, p. 1]. Un ejemplo cotidiano que podemos citar es la supervisión, reparación y mantenimiento de líneas de alta tensión. Ante estas necesidades, la comunidad científica ha planteado el uso de vehículos aéreos no tripulados, desarrollado para este objetivo complejas técnicas de control y avanzados sistemas de comunicaciones. Muy posiblemente las aplicaciones más importantes de estos vehículos las encontramos en el ámbito militar, esto debido a las últimas misiones bélicas desarrolladas a nivel mundial. Las labores más preponderantes han sido de vigilancia, recojo de datos e incursión en el espacio enemigo.

Uno de los inconvenientes más notorios del traspaso de la tecnología militar al entorno civil es la inviabilidad económica. Sin embargo, en la última década, una gran cantidad de universidades han desarrollado notables avances en el manejo de vehículos aéreos no tripulados. Inclusive, estos adelantos han sido motivo de competencias aéreas como es el caso de las organizadas por AUVSI (Association for Unmanned Vehicle Systems International). Para mayor información visitar <http://www.auvsi.org/>. No obstante, la realidad muestra que si bien es cierto se ha conseguido un alto grado de estabilidad y maniobrabilidad, la falta de estructuración del entorno, la limitada percepción sensorial y la imposibilidad del planificar a priori las tareas de inspección en sí, dificultan en total medida la aplicación de sistemas autónomos, requiriendo por ello, una intervención intermedia desde tierra a través de una interfase adecuada [6, p. 2].

Existen autores que presentan soluciones de vuelo autónomo para helicópteros.



Figura 1.1: Labores de mantenimiento e inspección de tendidos de alta tensión desde un vehículo aéreo. Para mayor información visitar <http://www.fkec.com/>.

Asimismo, existen sistemas de tele-operación disponibles utilizados en robots móviles submarinos; sin embargo, existen pocos sistemas que combinen los avances de los vehículos autónomos y de los vehículos tele-operados [6, p. 2]. El vacío existente no es abordado en gran medida por la presente tesis, pero es importante mencionarlo para que futuros trabajos se valgan de la presente investigación para resolver las dificultades ya antes mencionadas.

## 1.2. Personajes Relacionados

### 1.2.1. Leonhard Paul Euler

Leonard Paul Euler (ver figura 1.2) nació el 15 de abril de 1707 en Basilea, Suiza, y murió el 18 de septiembre de 1783 en San Petesburgo, Rusia. Fue un respetado matemático y físico, y es considerado uno de los más grandes matemáticos de la historia. Vivió en Rusia y Alemania la mayor parte de su vida y realizó importantes descubrimientos en áreas tan diversas como el cálculo o la teoría de grafos. También introdujo gran parte de la terminología y notación matemática moderna, particularmente para el área del análisis matemático, como

por ejemplo la noción de función matemática. También se le conoce por sus trabajos en los campos de la mecánica, óptica y astronomía. Se le considera como el principal matemático del siglo XVIII y como uno de los más grandes de todos los tiempos. También es uno de los más prolíficos, y se calcula que sus obras completas reunidas pueden ocupar entre 60 y 80 volúmenes. Una afirmación atribuida a Pierre-Simon Laplace expresa la influencia de Euler en los matemáticos posteriores: “Lean a Euler, lean a Euler, él es el maestro de todos nosotros”. En conmemoración suya, Euler ha aparecido en la serie sexta de los billetes de 10 francos suizos, así como en numerosos sellos postales tanto suizos como alemanes y rusos. El asteroide Euler recibió ese nombre en su honor.



Figura 1.2: Leonhard Paul Euler.

### 1.2.2. Aleksandr Mikhailovich Lyapunov

Nació el 6 de Junio de 1857 en Yaroslavl, Rusia y murió el 3 de Noviembre de 1918 en Odessa, Rusia, tras darse un tiro el día que su esposa murió. Lyapunov (ver figura 1.3) dejó una colección de estudios de gran importancia sobre el equilibrio de líquidos rotativos, la probabilidad y la estabilidad de sistemas dinámicos.

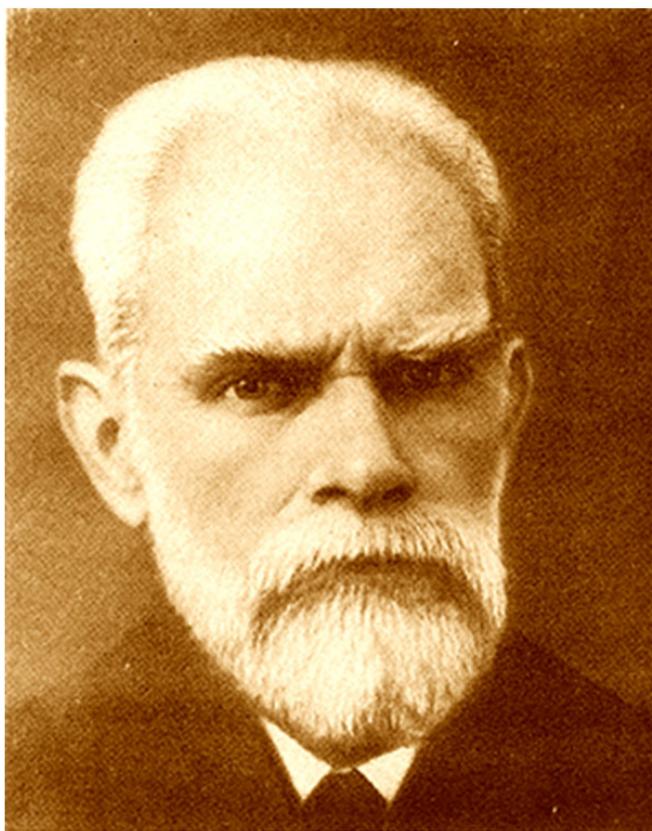


Figura 1.3: Aleksandr Mikhailovich Lyapunov.

Los métodos usuales de estudio de estabilidad no eran lo suficientemente buenos, ya que los errores debidos a la linealización hacían de ellos una aproximación no muy válida. Para salvar estas dificultades, Lyapunov desarrolló su propio concepto de estabilidad. Ideó métodos importantes de aproximación que proveen formas de determinar la estabilidad de sistemas de ecuaciones diferenciales ordinarias. Fue galardonado por sus contribuciones a varias academias co-

mo la Accademia dei Lincei (1909) y a La Academia de Ciencias (1916), Francia. También fue miembro honorario de la universidad de San Petersburgo, Kharkov y Kazan. A su muerte, muchos tributos y reconocimientos fueron ofrecidos. Por ejemplo el 6 de Junio de 1957, Sololev dio una conferencia del trabajo de Lyapunov en la “Teoría Potencial” en Moscú para un público compuesto y presidido por La Academia de Ciencias, La División de Ciencias Físicas y Técnicas de La Academia de Ciencias de la Universidad de Moscú, de la Sociedad Matemática de Moscú, del Instituto de Mecánica de la Academia de Ciencias, y del instituto de Automática y Telemecánica.

### 1.2.3. Igor Sikorsky

El ingeniero aeronáutico Ucraniano-Americano, constructor de dispositivos aéreos e inventor, Igor Sikorsky (1889-1972), mostrado en la figura 1.4, diseñó el famoso *clipper* volador y fue uno de los desarrolladores del helicóptero.

Igor Sirkorsky nació en Kiev, Ucrania, donde su padre fue profesor de psicología en la universidad St. Vladimir. Después de su graduación en la Academia Naval de St. Petersburgo, él estudió en París y en el instituto politécnico de Kiev. Estando en Alemania en 1908, escuchó de un dirigible que volaba construido por Count Zeppelin y luego retornó a París para estudiar aviación. A la edad de 20 años, Sikorsky construyó el primer helicóptero. Luego regresó a los aeroplanos, logrando, en 1910, el S-2 con un cierto éxito. En 1913, él construyó el primer aeroplano de 4 motores, “El Grande”, como se le llamó, fue adoptado por la armada Rusa y usado durante la Primer Guerra Mundial.

Sikorsky no simpatizaba con la Revolución Bolchevique de 1917 y en 1918, se dirigió a Paris, donde perdió todo su dinero tratando de construir un dispositivo volador. En 1919, arribó a los Estados Unidos, esperando encontrar una oportunidad más amplia para ejercitar sus talentos ingenieriles. Sus primeras



Figura 1.4: Igor Sikorsky (1889-1972). A su llegada a los Estados Unidos procedente de Rusia en 1919, más tarde dijera “*Encontré lo que había esperado, un país dinámico, enérgico y progresista.*”

experiencias fueros desalentadores, teniendo incluso la necesidad de apoyarse de las cátedras en una escuela para inmigrantes rusos. Luego en 1923, se unió a otros refugiados rusos para formar **Sikorsky Aero Engineering Company**. El compositor y pianista de concierto Sergei Rachmaninov fue el inversionista mayoritario y llegó a ser el vicepresidente de la nueva firma.

Dentro de algunos años la firma llegó a ser parte del gigante **United Aircraft and Transport Company**. Sikorsky y sus compañeros fueron excelentes ingenieros pero muy inexpertos hombres de negocios. En 1928, ellos empezaron la construcción de un gran hidroavión para **Pan American Airways** y perdieron US\$1 millón en contratos. En 1929, Sikorsky vendió 56 “aerial yachts” a acaudalados suscriptores, la mayoría de ellos envueltos en la bolsa de valores.

Cuando llegó la crisis del 1929, todos ellos quebraron y Sikorsky se encontró en una difícil situación financiera.

**United Aircraft** continuó respaldando a Sikorsky en la construcción de hidroaviones, lo cual culminó con la construcción del S-42, mostrado en la figura 1.5, para **Pan American**. Estos fueron extraordinarios logros en la ingeniería, pero todavía habían pérdidas en la compañía. Posteriormente, su construcción fue abandonada en 1938. **United**, sin embargo, consiguió que Sikorsky retornara a sus tempranos trabajos en helicópteros.



Figura 1.5: Hidroavión S-26 de Sikorsky.

El helicóptero había sido considerado como teóricamente trabajable, y entre 1920 y 1930, una gran cantidad de personas experimentaron con sus propios diseños. En 1939, Sikorsky evaluó su VS-300, su primer helicóptero práctico. Su éxito se debió al uso de un rotor que cambiaba el *Pitch* (ángulo de ataque de las hélices) de acuerdo a la dirección de vuelo. Durante la Segunda Guerra Mundial, los helicópteros no fueron usados hasta 1944, y sus contribuciones reales todavía hasta 1950, cuando Sikorsky se coronó fue uno de los líderes manufactureros del campo.

#### 1.2.4. Rudolf Emil Kalman

Fue un teórico matemático (ver figura 1.6) americano-húngaro y profesor emérito en el Instituto Federal Suizo de Tecnología. Fue mundialmente conocido por su co-invencción del filtro Kalman el cual es muy usado en los sistemas de control y aviónica actuales. Rudolf Kalman nació en 1930 en Budapest, Hungría. Obtuvo su grado de bachiller en 1953 y su licenciatura en 1954 en la escuela de Ingeniería Eléctrica del MIT (Massachusetts Institute of Technology). Su doctorado lo obtuvo en 1957 en la Universidad de Columbia. Se casó con Constantina Née Stavrou y tuvieron dos hijos: Andrew y Elisabeth.



Figura 1.6: Rudolf Emil Kalman.

Kalman ha trabajado como matemático investigador en el instituto de investigación para estudios avanzados en Baltimore desde 1958 hasta 1968. Ha sido profesor de la Universidad de Stanford desde 1964 hasta 1971. Es miembro

de la Academia de Ciencias de los Estados Unidos, la Academia Americana de Artes y Ciencias. También es un miembro extranjero en las Academias de Ciencias Húngara, Francesa, y Rusa. Recibió la medalla de honor de la IEEE en 1974, el premio de herencia de Richard E. Bellman en 1977, y el premio NAE Charles Stark el 2008.

### 1.2.5. Gareth Padfield

Atraído por la oportunidad de desarrollar el programa de investigación aeroespacial de vanguardia, El profesor Gareth Padfield (ver figura 1.7) tomó el lugar de James Bibby en Ingeniería Aeroespacial en la universidad de Liverpool en 1999. Pasó la mayor parte de su carrera investigando la teoría y la práctica del vuelo para aeroplanos y helicópteros en “La Agencia de Investigación de Defensa y Evaluación”, donde él llegó a ser el científico principal y Asociado mayor de la DERA.



Figura 1.7: Gareth Padfield.

En Agosto de 2004, Gareth fue nombrado jefe del Departamento de Ingeniería (Incluyendo Aeroespacio, Mecánica, Ciencia de los Materiales, Manufactura, Disciplinas de Diseño y Civiles) lo cual comprendía una comunidad de 200 a 1000 miembros y tenía un portafolio evaluado en 20 millones de libras esterlinas. Como cabeza de la Ingeniería, su visión fue y es ayudar a crear entornos donde la calidad de investigación y enseñanza sean reconocidas internacionalmente como líderes mundiales.

Los intereses de Gareth incluyen calidad de maniobra de vuelo y control, modelado, simulación y dinámica de vuelo de aeroplanos y helicópteros. Los proyectos actuales bajo su supervisión incluyen: desarrollo de calidades de maniobra y aliviamiento de carga-par. Gareth también ha publicado más de 150 **journals**, **papers** de conferencias y reportes técnicos de sus descubrimientos de investigación. La segunda edición de su libro **Helicopter Flight Dynamics** fue publicada en junio de 2007.

### 1.2.6. Bernard Mettler

Bernard Mettler (ver figura 1.8) ha tenido como intereses principales el área de control y guía de vehículos autónomos. Un problema importante para los vehículos aéreos no tripulados es el navegar autónomamente dentro de ambientes complejos o desempeñar tareas que requieran de comportamientos dinámicos complejos. En contraste con muchos robots móviles, los vehículos aéreos pueden presentar un ancho rango de comportamientos dinámicos, pero a la vez son susceptibles a condiciones atmosféricas; esta característica es acentuada cuando el vehículo disminuye de tamaño. Los sistemas autónomos requerirán una fuerte integración entre planeamiento y control; por ejemplo, son capaces de llevar el control de la dinámica, mientras se realiza el planeamiento de trayectorias. El profesor Mettler se ha enfocado en el modelado matemático, sistemas de control

de fuselaje completo, y eficiencia en los algoritmos de planeamiento de trayectorias computacionales, con énfasis en los sistemas de integración.



Figura 1.8: Bernard Mettler.

## 1.3. Estado del Arte

### 1.3.1. Definición de vehículos aéreos no tripulados (UAV)

Siguiendo la clasificación de la Asociación Europea de Vehículos no Tripulados: **EURO-UVS** (<http://www.euro-uvs.org/>), un UAV puede definirse como un “vehículo aéreo motorizado,” el cual puede ser [6, p. 6]:

- Reutilizable o desechable,
- remotamente controlado, semi-autónomo, autónomo o cualquier combinación de estas tres características,

- capaz de sobrevolar un objetivo o
- capaz de transportar diferentes tipos de cargas, haciéndoles capaces de realizar tareas útiles en la atmósfera terrestre o más allá de la misma, y con una duración de vuelo acorde a la tarea a realizar.

### **1.3.2. Antecedentes históricos**

En esta sección se muestran algunos de los avances desde tiempos muy remotos hasta tiempos modernos en lo referente a vehículos aéreos. La información que se muestra a continuación es una recopilación de tesis desarrolladas referente al tema y de algunas páginas electrónicas disponibles.

En tiempos modernos, los UAV aparecieron en la Primera Guerra Mundial; sin embargo, la idea de una máquina voladora fue concebida todavía 2500 años atrás [21, p. 15].

#### **1.3.2.1. Diseños incipientes**

Ha sido documentado que la mayor contribución en los mecanismos autónomos ocurrió durante la era de Pitágoras, quien fue el alumno de Thales por algunos años. El primer gran avance en los mecanismos autónomos es atribuido a Arquitas de Tarento quien vivió al sur de la Magna Grecia (ahora Italia). Él no fue solo el número 1 en las matemáticas, sino que también fue ingeniero. Aplicando nociones geométricas y observaciones del funcionamiento de las estructuras, él diseñaba mecanismos. Por el año 425 A.C., creó el primer UAV conocido también como la paloma voladora. Dicho mecanismo podía volar moviendo sus alas, la energía era conseguida desde un mecanismo al interior de su estómago (ver figura 1.9). El mecanismo podía volar por cerca de 200 m, y no podía volver al volar a menos que el mecanismo fuera reanudado. Como fuera descrito por el escritor latino Aulus Gellius, alegando que este aparato se propulsaba utilizando chorros de vapor de

agua [21, p. 16].



Figura 1.9: Ilustración de un artista de la paloma voladora, el primer UAV documentado de la historia. Se ha reportado que voló por cerca de 200 m.

Durante la misma era de la matemáticas pitagóricas, en otra parte del mundo antiguo, en lejana China, cerca del 400 A.C., los chinos documentaron la idea de un dispositivo de vuelo vertical (ver figura 1.10). El mecanismo consistía de plumas a un lado de un palo. El mecanismo era agitado por las manos lo suficiente rápido para generar un empuje que pudiera ser lanzado en vuelo libre [21, p. 16]. Este sencillo, pero ingenioso mecanismo guardaba cierta semejanza con un resorte de arco. Launoy y Bienveu mostraron el ingenioso aparato en una reunión de la Academia Francesa de las Ciencias el 28 d abril de 1784; el mecanismo se elevó rápida y limpiamente hasta chocar contra el techo y caer después al suelo. En aquellos momentos, ni los inventores ni los sabios allí congregados se dieron cuenta que el sorprendente aparato había superado dos de las barreras fundamentales en el desarrollo del helicóptero: tenía una fuente propia de energía y sus hélices contra-giratorias eliminaban el problema del par [23, p. 18].

Más tarde, diecisiete siglos después, de manera similar a lo concebido por



Figura 1.10: Representación mecánica de un objeto volador. El eternamente juguete entraña principios en los que se bases el funcionamiento de todos los helicóptero modernos.

Arquitas de Tarento, un ave voladora hecha por un desconocido ingeniero del Renacimiento fue reportada (ver figura 1.11). No se comprobó que este diseño se basara en el de Arquitas, pero el concepto era similar.

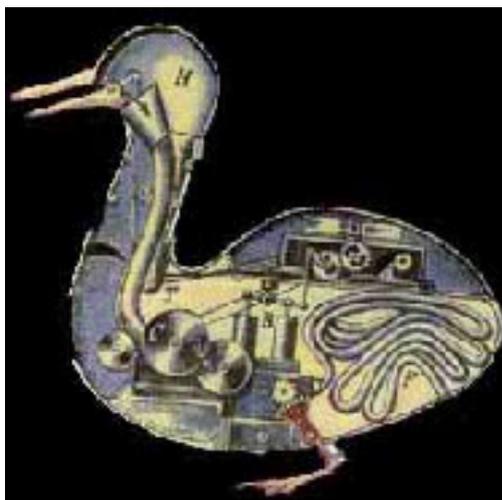


Figura 1.11: Ave voladora con un mecanismo en el estómago, atribuida a un ingeniero renacentista.

Numerosos teóricos y experimentadores de la aeronáutica continuaban persiguiendo el sueño de poder crear una máquina de vuelo vertical. Leonardo da Vinci, en 1483, diseñó un mecanismo aéreo capaz de suspenderse. Llamado también propulsor aéreo (mostrado en la figura 1.12) tuvo 5 m de diámetro y la idea fue hacer que el eje gire y que este tenga la suficiente energía de suministro. Para algunos expertos esta máquina es considerada como el antecesor del helicóptero [21, p. 17].

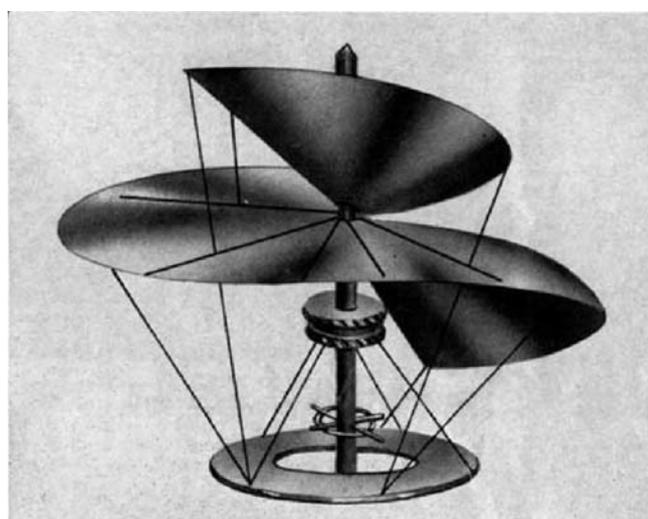


Figura 1.12: Propulsor de aire de Leonardo da Vinci. Un plano continuo en espiral impartía fuerza ascensional al aparato; la energía procedía del un sencillo mecanismo de relojería. Si el mecanismo tuvo éxito, como dejan traslucir las notas de Leonardo, fue el primer mecanismo aéreo que voló con una fuente de energía incorporada en sí mismo.

Dos diseños fueron ideados a raíz del modelo chino, los cuales fueron documentados en 1754 y 1783. El primer crédito fue para Mikhail Lomonosov quien diseñó un rotor coaxial potenciado por un dispositivo de resorte. El segundo fue atribuido a Launoy y Bienvenue quienes construyeron hélices contra-giratorias utilizando plumas de pavo o seda. En 1843, Sir George Cayley diseñó un convertiplano (ver figura 1.13) capaz de suspenderse. Él creía que la energía proveniente de plantas podría ser apropiada para los motores a vapor; sin embargo, esta todavía no había sido posible de ser usada para impulsar el vuelo aéreo [21, p. 18].

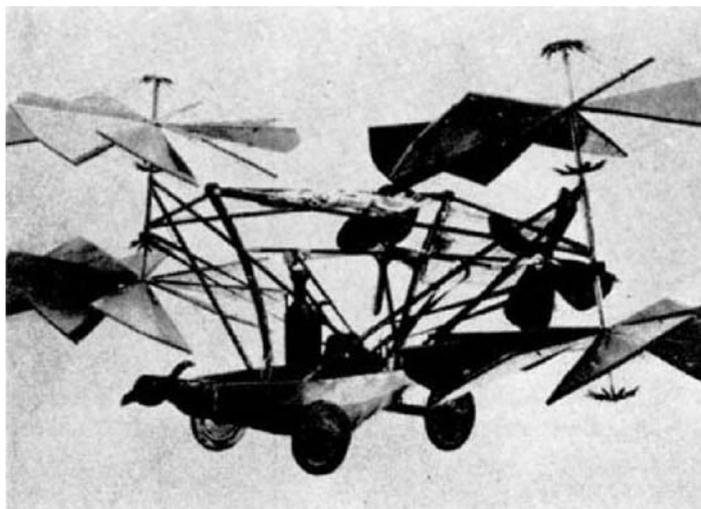


Figura 1.13: Carruaje aéreo de George Cayley.

Una máquina de vuelo vertical fue diseñada por W. H. Phillips. Un caldero en miniatura fue utilizado para producir el vapor que salía expulsado por unas boquillas sobre las palas. Lamentablemente, su mecanismo era ingobernable; cuando lo lanzó por primera y única vez, se elevó rápidamente en el aire, pasó rápidamente por encima de los campos cultivados y desapareció [23, p. 19].

Para la segunda mitad del siglo XIX, las muy celebradas ascensiones en globo y los intentos de vuelo habían disparado la imaginación (ver figura 1.14) del público con sueños de aeronaves mecánicas. Los ejemplos de helicópteros imaginarios que vuelan van desde el más caprichoso hasta el más decidido diseño como es el caso de Gabriel de La Landelle (ver figura 1.15).

No obstante, Gustave de Ponton d'Amécourt en 1860 logró volar un pequeño helicóptero impulsado por vapor, mostrado en la figura 1.16. Fue en ese tiempo que el término helicóptero fue acuñado basado en la palabra griega compuesta *ελικοπτερο*. La primera parte *ελικας* que significa hélice y la segunda parte *πτερου* que significa pluma o ala.

En 1880, Thomas Alva Edison experimentó diferentes configuraciones de rotor, usando al final un motor eléctrico para la potencia. Este nuevo diseño



Figura 1.14: Una irónica visión del siglo XX nos muestra a unos respetables ciudadanos paseando plácidamente por el aire portando rotores sujetos a la espalda con correas.

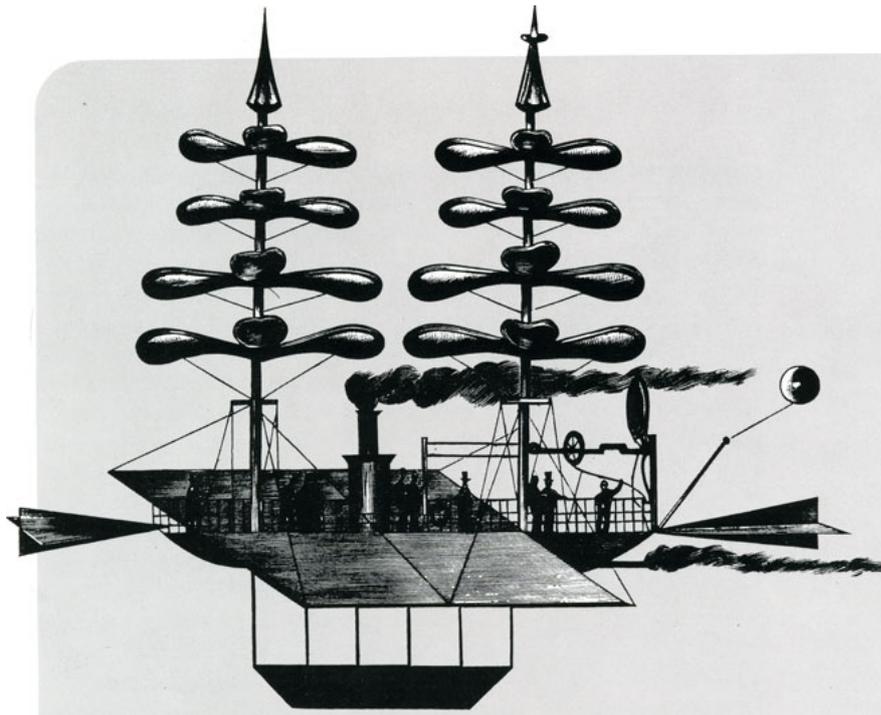


Figura 1.15: Gabriel de La Landelle ideó en 1863 esta “nave aérea de vapor” que usaba ocho rotores elevadores y dos mástiles, cada uno de ellos coronado por un paracaídas de paraguas. También tenía dos grandes alas, un hélice a popa y una góndola colgante para los pasajeros.

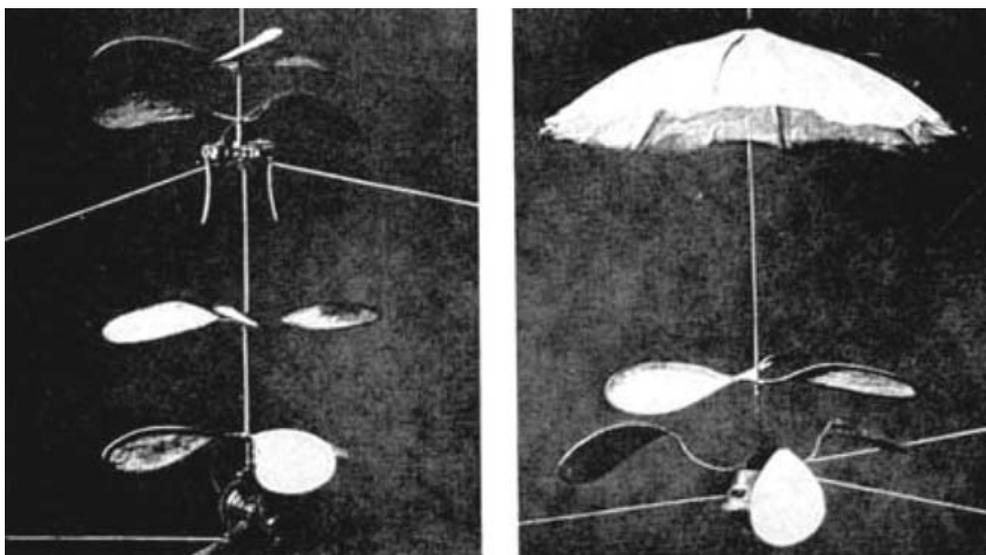


Figura 1.16: Helicóptero de Ponton d'Amécourt.

mostró las mejores habilidades de suspensión y requirió grandes hélices para su construcción. Después de un accidente que tuviera con un modelo de helicóptero que usaba piroxilina, donde uno de sus ayudantes salió gravemente lastimado, Edison abandonó calladamente su investigación activa en el vuelo vertical. No obstante, él nunca perdió la convicción de que el helicóptero era el vehículo definitivo para circular en el aire. *“Cualesquiera que sean los progresos que pueda hacer el aeroplano -insistía años mas tarde-, el helicóptero llegará a ser el tema favorito de los estudios avanzados de la aeronáutica”* [23, p. 25].

En 1907, Louis Breguet ideó un giro-plano el cual logró suspenderse por primera vez con una persona en su interior. El aparato pareció ingobernable por el piloto, pero el aparente éxito hizo que Breguet retornara a su mesa de dibujo. Mientras tanto en la ciudad de Lisieux, otro ingeniero francés, Paul Cornu, estaba trabajando en un diseño propio (mostrado en la figura 1.17) para una máquina de ascenso vertical. Paul Cornu diseñó un dispositivo aéreo de dos rotores, el cual logró elevar a un hombre. Los rotores giraban en sentidos contrarios, y la máquina logró volar por escasamente 20 segundos hasta que cayera estrepitosamente sobre

el suelo [21, p. 19].



Figura 1.17: Helicóptero de Paul Cornu.

Uno de los grandes avances en la era moderna del helicóptero fue logrado por Igor Sikorsky, pese a que su modelo fue construido en 1909, este (mostrado en la figura 1.18) no pudo volar por problemas de vibración y por falta de potencia en el motor [21, p. 20]. Por supuesto es importante recalcar que en paralelo al avance de aeronaves de despegue vertical, los aeroplanos habían empezado a evolucionar con un siglo más de anticipación, teniendo su primer vuelo exitoso con los hermanos Wright en 1903.

### 1.3.2.2. Diseños póstumos a la Primer Guerra Mundial

Los UAV entraron en funcionamiento durante la Primera Guerra Mundial, estos modelos se ven las figuras 1.19, 1.20, 1.21 y 1.22.

Sin embargo, fue Sikorsky quien construyó el clásico helicóptero (1939) mostrado en la figura 1.23. La aeronave podía mantenerse en el aire durante 15 minutos, pero este modelo podía volar en todas direcciones salvo hacia adelante. El dijo entonces “*Ese es uno de los pequeños problemas de ingeniería que todavía*

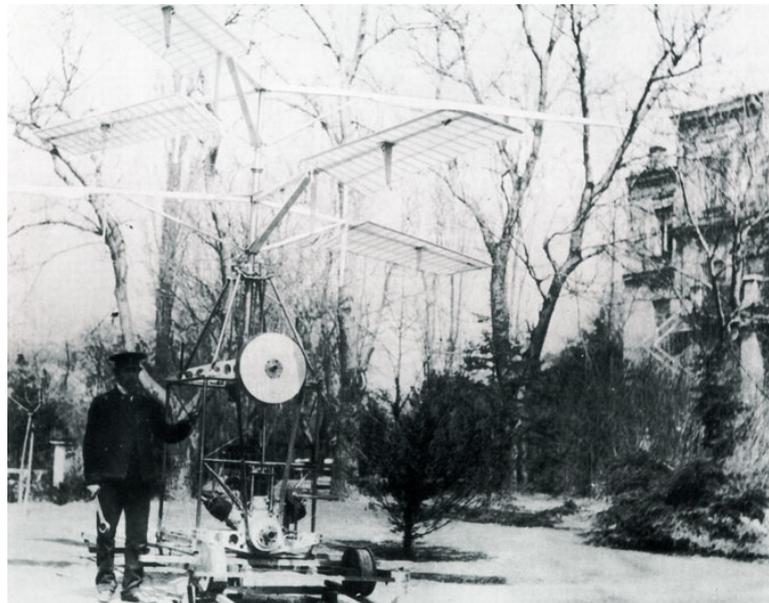


Figura 1.18: Frente a la casa de su padre en Kiev, en 1910, el joven Sikorsky se apresta a hacer algunos ajustes a su segundo helicóptero. Reconstruido a partir de un modelo de 1909 el cual fuera inviable por sus muchas vibraciones, el nuevo modelo pesaba menos, y en palabras del propio Sikorsky, se parecía a “*una enorme mariposa*”. Podía elevar su propio peso (181 Kg) pero nada más: ni combustible para sostener el vuelo ni piloto para gobernarlo.

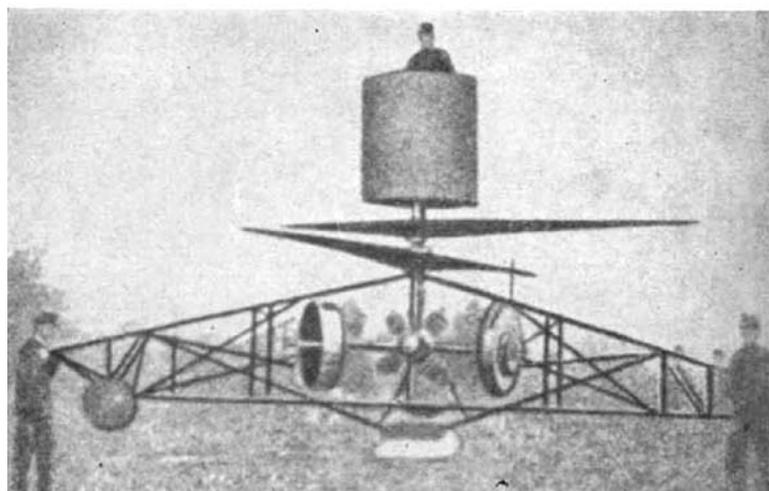


Figura 1.19: El helicóptero de Stephan Petroczy y Theodore Von Karman consistente de dos propulsores sustentadores superpuestos.

*no hemos resuelto*”. Los avances continuaron con Sikorsky liderando el mercado hasta la construcción del primer helicóptero de transporte comercial en 1950,



Figura 1.20: El helicóptero de Bothezat con seis rotores de palas (1922) fue respaldado por la armada estadounidense.



Figura 1.21: El autogiro de Juan de la Cierva fue un dispositivo híbrido con alas y cola, pero también con un rotor vertical sobre el fuselaje.

llamado S-55 Chickasaw (H-19), como se aprecia en la figura 1.24 [21, p. 22].  
 Novedosos diseños e importantes logros se presentaron por los siguientes treinta años, incluyendo nuevos diseños en el rotor, los cuales se verán en la siguiente sección.



Figura 1.22: El helicóptero Fa-61 (1936) fue una máquina de dos rotores de lado a lado. Este es el primer helicóptero que demostró un completo control de vuelo y exitosas auto-rotaciones.

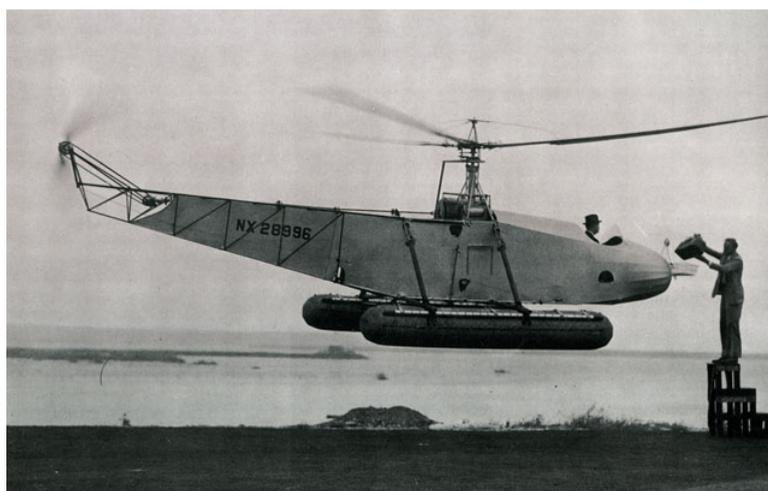


Figura 1.23: Cerniéndose tan abajo y tan inmóvil que un ayudante podía colocar un paquete en un canastilla acoplado a la proa de la aeronave, Sikorsky demuestra la maniobrabilidad de su VS-300. Los pontones acoplados a los patines del helicóptero hacían que pudiera utilizarse tanto desde tierra como desde agua.

### 1.3.2.3. La nueva era de los UAV

Sin duda lo que hoy conocemos de los UAV se formó en el seno del departamento de defensa norteamericano. Los primeros diseños se hicieron en los años 50 y se siguieron usando en combates bélicos hasta los 60. En la actualidad existen 30 modelos de vehículos diferentes reportados [6, p. 6]. Uno



Figura 1.24: Al colocar el motor en la proa del S-55, un helicóptero que se usó mucho en Korea, Sikorsky consiguió espacio para una cabina con capacidad para 10 personas. El motor se conectaba al rotor por medio de un árbol de transmisión inclinado que pasaba entre los asientos de una carlinga. El aparato que se reproduce aquí luce los colores de la Marina Real canadiense que utilizó el S-55 para misiones de rescate.

de los pioneros fue el AM-34N Firebee, mostrado en la figura 1.25 (<http://www.northropgrumman.com/>). Con una envergadura de 4 m y una longitud total de 7 m, este realizaba tareas de reconocimiento y toma de datos. Logró también una altura máxima de 18000 m y un alcance de 3800 Km [6, p. 6].

A principios de los 70, apareció el “154 Compas Arrow” capaz de volar a gran altitud y de ser difícil de detectar. La navegación y el guiado se realizaban usando una unidad inercial. A finales de los 70, empezó el proyecto “Aquila” con un costo inicial de US\$ 123 millones. Su objetivo era ser ligero y obtener información más allá de lo que la vista pudiera observar. Funcionó en 7 de sus 105 pruebas [6, p. 6].

Basado en el “Firebee” y en capacidades de un vehículo autónomo, se desarrolló el “AQM-34L” conocido como el Vehículo Pilotado Remotamente o Remoted Piloted Vehicle (RPV). Este era lanzado desde un avión mayor y



Figura 1.25: Firebee fabricado por NORTHROP GRUMMAN.



Figura 1.26: Compass COPE actualmente retirado al museo de la USAF (United States Air Force).

fue utilizado para realizar vuelos fotográficos. Posteriormente a mediados de los 70, se desarrolló el “COMPAS COPE” mostrado en la figura 1.26 (<http://www.nationalmuseum.af.mil/factsheets/factsheet.asp?id=607>), el cual a diferencia de los otros RPV, era capaz de despegar y aterrizar. Además, batió récord al permanecer en vuelo por 28 horas y 11 minutos sin reposar [6, p. 7].



Figura 1.27: Dragon Drone usado por la U.S. Army (Armada de los Estados Unidos) y por la Marin Corps (Infantería de Marina).

El “Dragon Drone” mostrado en la figura 1.27 (<http://www.designation-systems.net/dusrm/m-147.html>) que era capaz de ser lanzado con un sistema neumático, podía ser operado hasta una cierta altura de trabajo y desde allí se desempeñaba de manera autónoma. Estaba hecho para labores de reconocimiento y era recuperado por medio de un paracaídas. El “BQM-145 A Peregrine” fue el primer vehículo de medio alcance de dispuso de una unidad de control embarcada (On Board Mission Logic Control Unit). El control se apoyaba de un sistema inercial con GPS, el sistema era recogido por un paracaídas en tierra o mar [6, p. 7].

A pedido de la Marina de Guerra de los Estados Unidos se empezó a construir vehículos de despegue y aterrizaje vertical como el “Eagle Eye” (VTOL), mostrado en la figura 1.28 (<http://www.bellhelicopter.com/en/aircraft/military/bellEagleEye.cfm>). Los primeros vuelos fueron sólo de suspensión



Figura 1.28: Eagle Eye de Bell Helicopter.

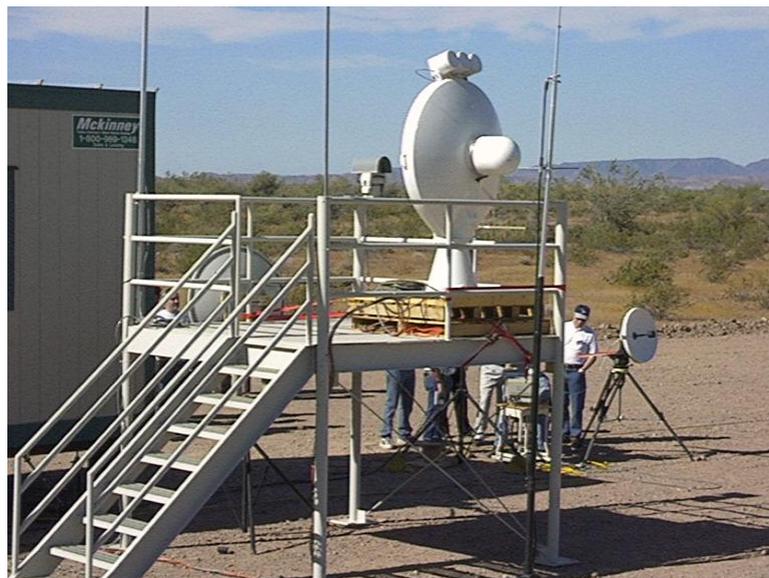


Figura 1.29: Antenas de seguimiento del Eagle Eye.

y se realizaron por el año 1992. No realizó despegues ni aterrizajes de manera automática hasta 1998, utilizó una antena móvil (mostrada en la figura 1.29) para una mejor transmisión. Posteriormente a finales de los años 90, los canadienses crearon el “CL-227 Sentinel” (mostrado en la figura 1.30) con capacidad de despegue y aterrizaje vertical. Contaba con una capacidad de carga de 100 Kg, una

autonomía de 6 horas y un alcance de 200 Km. Actualmente se está desarrollando el “Phampton Sentinel” (mostrado en la figura 1.31) que está basado en el funcionamiento del boomerang y dotado con cámaras. Este dispositivo es capaz de ser lanzado de manera manual [6, p. 8].



Figura 1.30: CL-227 Sentinel de la Fuerza Aérea de Canadá.



Figura 1.31: Se muestra el Phampton Sentinel cuando es lanzado por un soldado.

A comienzos de siglo XXI, los máximos exponentes en UAVs son el Global Hawk de largo alcance (mostrado en la figura 1.32, <http://www.is.northropgrumman.com/>) y el Predator de medio alcance (mostrado en la figura 1.33, <http://www.airforce-technology.com/projects/predator/>) el cual empezó a estar operativo el año 2000 [6, p. 8].



Figura 1.32: Global Hawk de NORTHROP GRUMMAN.



Figura 1.33: Predator de la USAF.

### 1.3.3. Trabajos relacionados

En el campo civil, diversas universidades de los Estados Unidos y Europa lograron grandes avances en el desarrollo de helicópteros autónomos. También, han tenido gran repercusión los trabajos hechos por el profesor M. Sugeno en el control del mini-helicóptero RMAX (mostrado en la figura 1.34, <http://www.yamaha-motor.co.jp/global/news/2002/02/06/sky.html>) adquirido de la compañía Yamaha la cual comercializa varios de estos modelos debido a su pequeño tamaño y versatilidad. El método de Sugeno de guiado es a través de puntos de paso u órdenes manuales (de un mouse o un joystick) en tres dimensiones [6, p. 8].



Figura 1.34: RMAX de Yamaha el cual es muy usado en proyectos de UAVs.

#### 1.3.3.1. Reseña histórica del desarrollo del mini-helicóptero autónomo RMAX-50 (figura 1.35) de la Universidad Carnegie Mellon

Uno de los prototipos no tripulados propuestos fue el que se desarrolló en la Universidad Carnegie Mellon (CMU). Dicho helicóptero fue hecho en base a un modelo proveído por la marca Yamaha. Este tuvo un control de posición en el espacio, y las tareas comunes que se pudieron implementar en él fueron las

pruebas de seguimiento de trayectorias definidas, pudiendo seguir trayectorias cuadradas y circulares. El control de posición se realizó en base al manejo de un GPS para la adquisición de datos de posición global.

Además, una de las características más relevantes que se implementó en este prototipo fue el sistema autónomo de manejo el cual permitía que el helicóptero puede desplazarse por zonas no exploradas anteriormente, enviando en todo momento señales de posición espacial. Esta característica permitió darle un sinnúmero de usos en ámbitos donde se creía fuera muy complicado para el ser humano. Por ejemplo, las aplicaciones más directas que se le dio fueron el tomar fotografías de volcanes en actividad, y el manejar cámaras para televisar desde ángulos que no son accesibles para el hombre; también, se le ha propuesto para la detección de fallas en represas, en oleoductos y en zonas externas de las embarcaciones marinas.

Para dar cuenta de los avances que ha tenido este proyecto en la CMU, citaremos sus logros cronológicamente.

**Septiembre de 1991** La CMU inició el estudio del control de altitud, además de desarrollar un modelo eléctrico montado sobre una plataforma-brazo. Este prototipo contaba con un encoder óptico (sensor de posición) para la medición de ángulos de actitud (roll, pitch y roll) en tiempo real.

### **Febrero de 1992**

- Se construye el primer modelo de vuelo libre de seis grados de libertad, desarrollado para evaluar varias estimaciones de posición y control de sistemas.
- Se implementa un modelo eléctrico anexo a polos de grafito para su seguridad y estimación de posición para un buen aterrizaje.

- Composición de material ligero y diseño de vigas de aire sin fricción lo cual permitió un vuelo libre en un área con forma de cono.
- Paradas mecánicas diseñadas para prevenir posibles choques o vuelos sin dirección.

### **Febrero de 1994**

- Primera plataforma autónoma.
- Implementado con sistema de visión, sistema de control, plataforma de sensores, sistema de potencia, interferencia de RF y sobre todo un sistema integrado.
- Permite 1.5 m de vuelo longitudinal sobre el piso.
- Límite en el vuelo lateral y vertical.
- El helicóptero es atado con fajas las cuales son ajustadas a tierra y a dos polos posicionados a cada lado de la plataforma.

### **Julio de 1998**

- Sistema de mapeo por láser a bordo.
- Se maniobró sobre el cráter Haughton, Devon Island NWT, Canadá.

### **Agosto de 2006**

- Sistema de control para un despegue y aterrizaje autónomo, además de un seguimiento suave de trayectorias.
- Sistema de ensayo de situaciones difíciles.
- Estimador difuso de datos desde el GPS receptor, 3 ejes angulares y sensores inerciales, además de un índice de visión basado en odometría.

- Diseño típico de visión capaz del detectado de posición, seguimiento de objetos múltiples, la discriminación de color, y mapeo aéreo.
- Sistema de estabilización por cámara.
- Láser 3D para el escaneo.



Figura 1.35: Equipo de la universidad Carnegie Mellon con el RMAX de Yamaha.

### **1.3.3.2. Asociación Internacional para Vehículos Aéreos AUVSI (Association for Unmanned Vehicle Systems International)**

Competencia existente desde 1991. Desde 1991 al 1995, la prueba consistía en recoger un disco. Luego, entre 1996 y 2000 la prueba trato del reconocimiento de victimas en un desastre tóxico. Actualmente y desde el 2001, la competencia consiste en 4 niveles a desarrollar:

- Vuelo autónomo de 3 Km a través de puntos predeterminados.

- Identificación de un edificio a través de un símbolo en particular e identificación de la ventana para ingresar.
- Ingreso en el edificio seleccionado, navegación y captura de imágenes internas.
- El último nivel consiste en realizar todas las pruebas en menos de 15 min.

Los hitos de la competencia se describen a continuación:

- 1993, primer despegue y aterrizaje autónomo (Georgia Tech).
- 1995, misión de recuperación de un disco completada (Stanford University).
- 1997, misión de inspección de basura tóxica completada (Carnegie Mellon University).
- 2000, misión de búsqueda y rescate en un gran desastre (Universidad Tecnológica de Berlin).
- 2001, navegación sobre 3 Km de líneas guía completada (Georgia Tech).
- 2002, nivel 2 casi completado. Los edificios y ventanas fueron correctamente mapeados (Georgia Tech).
- 2003, nivel 2 completado. El edificio correcto fue identificado usando el símbolo guía. El portal abierto fue encontrado (Georgia Tech).

#### **1.3.3.3. Antecedentes del vuelo de mini-vehículos aéreos en interiores**

Recientemente, muchos grupos de investigación han estado progresivamente desarrollando vehículos aéreos autónomos. La evolución de robots voladores en miniatura ha sido posible gracias al reciente crecimiento del interés militar y civil en vehículos aéreos no tripulados. Mientras muchos de los trabajos se han enfocado en ambientes exteriores, una especial atención han tenido los vehículos

aéreos de pequeña escala que operan en interiores. Estos dispositivos operan en zonas no estructuradas y desconocidas. Muchos de ellos son usados para exploración, localización de objetos y búsqueda. También se vislumbra a estas máquinas voladoras operando en salvamento y rescate.

Dos son las ventajas principales de los vehículos aéreos con respecto a los vehículos terrestres: su movilidad y maniobrabilidad. Sin embargo, tanto los vehículos aéreos como los terrestres tienen como objetivo principal alcanzar una posición deseada sin la intervención del hombre, pero también ambos experimentan el problema de estimación de sus estados y planeamiento de movimiento. Estos problemas han sido resueltos con acierto, pero sólo para vehículos terrestres. Pese a ello, ya existen algunas páginas electrónicas como [www.openquadrotor.org](http://www.openquadrotor.org) que ofrecen una plataforma para simulación de vehículos voladores y creación de prototipos, permitiendo, de esta forma, adaptar dicho sistema a diversos propósitos.

Con el objetivo de maniobrar los vehículos aéreos de manera segura en ambientes interiores, estos robots deben enfrentar los siguientes retos. Mientras los vehículos terrestres no tienen límites de peso, los vehículos aéreos sí están tremendamente limitados en peso. Por ejemplo, nuestro quadrotor tiene un peso de carga límite de 200 g. Otro problema es la odometría indirecta. Mientras los vehículos terrestres pueden estimar su posición directa y precisamente, los vehículos aéreos se valen de dobles integraciones de sus aceleraciones (detectadas por sensores inerciales) o de métodos de emparejamiento de barrido láser o Scan Matching. El problema de recursos computacionales limitados es también un gran reto que vencer. Y por si fuera poco, los helicópteros en general tienen rápidas y más complejas dinámicas de vuelo, lo que hace que su control sea complicado.

Muchos investigadores se han enfocado en el modelado y control de vehículos aéreos [2], [19]. Ellos han enfatizado el desarrollo de pequeños helicópteros como el CALIBER 5 o quadrotores. Entre los campos de investigación tenemos:

estabilización de actitud, despegue vertical, y aterrizaje.

Entre los reportes científicos que marcaron hito en el desarrollo de vehículos aéreos no tripulados en interiores tenemos Hoffmann et al. [9], quien muestra el diseño y desarrollo de un seguidor autónomo en miniatura de puntos preestablecidos, y la creación de una plataforma que permite la experimentación y validación de algoritmos de control multi-agente (ver figura 1.36). Grzonka et al. [7] presenta un sistema general el cual permite que un pequeño vehículo aéreo pueda operar en interiores. El sistema incluye un módulo de estimación de localización, estimación de actitud y altitud, y SLAM (Simultaneous Localization and Mapping-Localización, Mapeo y Localización Simultáneo). Además tiene un control del ángulo Yaw, el cual le permite evitar colisiones (ver figura 1.37). Achtelik et al. [1] propone una solución para permitir que un helicóptero quadrotor navegue, explore, y localice objetos de interés en un ambiente no estructurado y desconocido de manera autónoma. Este usa una odometría visual para reconstruir la posición del helicóptero en el espacio 3D, es decir, el movimiento del helicóptero fue computado dentro de los 6 grados de libertad (ver figura 1.38).

#### 1.3.3.4. Problema de control del vuelo en interiores

Para alcanzar y mantener una posición en un ambiente no explorado de manera autónoma, necesitamos estimar de manera precisa la pose ( $x$ ,  $y$  y orientación) del vehículo en todo instante. También, para mantener una cierta trayectoria con apropiada precisión, el robot necesita saber su posición en el ambiente en cualquier punto de dicha trayectoria. Este proceso se conoce como *Localización*. Seguidamente, para computar la trayectoria, el mapa del ambiente debe de ser conocido y obtenido mientras el vehículo viene operando. Este proceso es conocido como SLAM. Finalmente, el sistema de control tiene que mover el vehículo aéreo a través de la trayectoria deseada basado en la pose estimada a

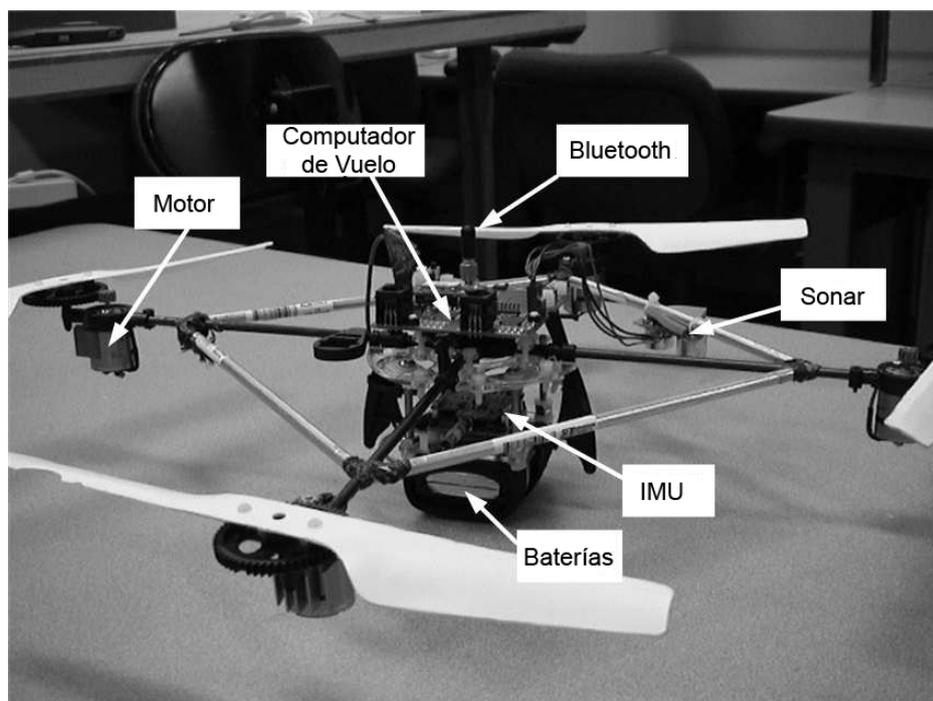


Figura 1.36: Configuración del sistema STARMAC de la Universidad de Stanford.

través de los sensores que se encuentran a bordo.

Muchos autores han resuelto el problema de control por medio de la incorporación de GPSs; sin embargo, en nuestro caso no tenemos este beneficio, ya que en interiores los GPSs no funcionan. Además, la precisión de los GPS no es suficiente en ambientes interiores. Por eso, el sistema de estimación de pose debe de estar basado en información de los obstáculos que se encuentra en el mismo ambiente, es decir, el sistema a bordo debe estar compuesto de un sensor láser y un IMU (Inertial Measurement Unit), quienes son los que dan información de un medio o contorno próximo e información inercial respectivamente. Finalmente, por medio el uso de el muy conocido algoritmo de Scan Matching o Emparejamiento de Barrido, la pose de un helicóptero quadrotor podrá ser estimada.

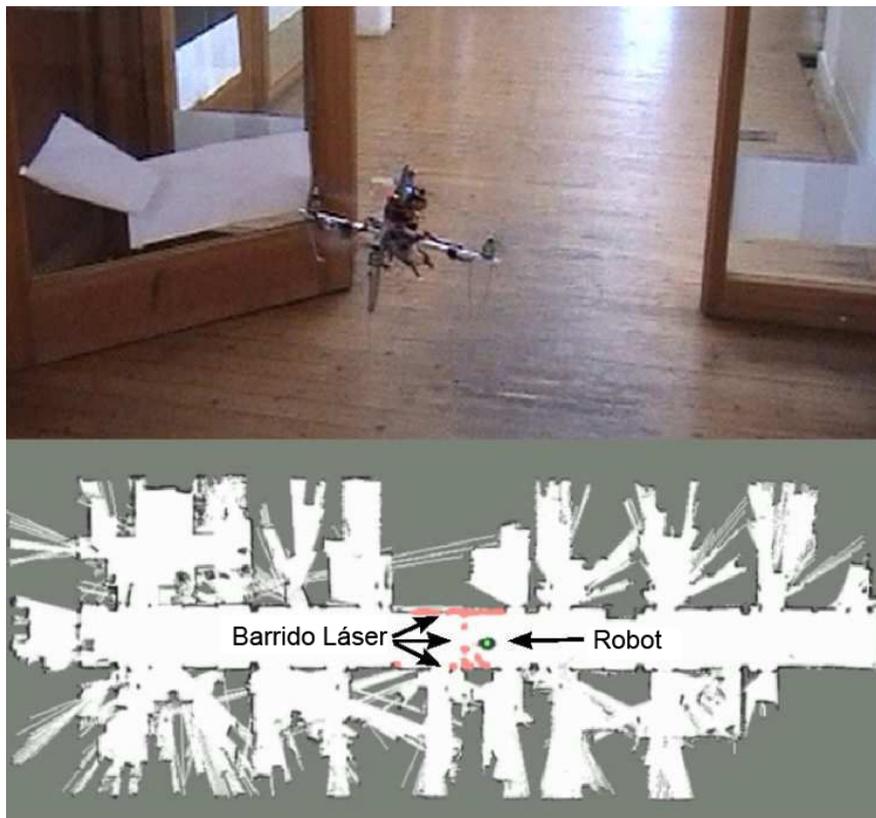


Figura 1.37: El quadrotor equipado con un sistema de navegación durante una misión (arriba). La posición estimada on-line durante el vuelo (abajo).

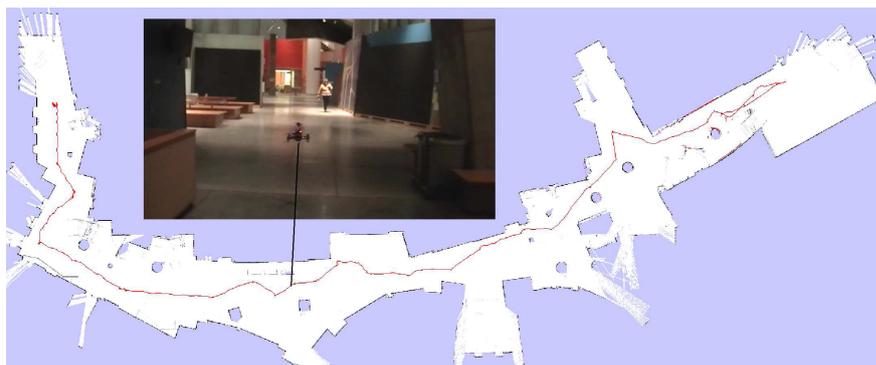


Figura 1.38: Helicóptero quadrotor realizando SLAM.

#### 1.3.4. Aplicaciones de los vehículos no tripulados

A continuación vamos a detallar las aplicaciones más importantes de los UAVs, así como también de los vehículos totalmente autónomos bajo cualquier condición

climatológica.

#### **1.3.4.1. Misiones Militares**

Una de las grandes aplicaciones de los vehículos no tripulados o autónomos sin duda se encuentra en el ámbito militar. Gracias al pequeño tamaño que pueden llegar a tener, éstos son idóneos para evitar ser detectados por los radares del enemigo en tareas de espionaje [6, p. 13]. En este terreno, es posible encontrar desde mini-UAVs hasta aerosondas. Como vehículos telecontrolados, se están utilizando en la monitorización de fronteras y en el detección de minas antipersonales. En el ámbito de vigilancia, se destaca el amplio rango de áreas observables, la forma periódica y repetitiva con que pueden realizarse estas observaciones.

#### **1.3.4.2. Inspecciones**

La capacidad de vuelo estático y la gran maniobrabilidad permite que los helicópteros puedan realizar inspecciones en puentes, edificios, presas, y líneas de alta tensión (ver figura 1.39). Estos vehículos dotados de cámaras podrán permitirán hacer inspección en lugares de difícil acceso en corto tiempo. Habitualmente teleoperados, éstos pueden inspeccionar materiales peligrosos por medio de tomas aéreas que serán posteriormente enviadas a expertos.

#### **1.3.4.3. Búsqueda y rescate**

Intrínseco a su definición éstos vehículos pueden sobrevolar rápidamente zona de desastres reconociendo (ver figura 1.40) personas que pudieran estar en peligro. Incluso podrían determinar la posición de las víctimas y así poder guiar a los rescatistas a la zona e siniestro. Además se pueden transportar en cualquier condición climatológica lo que podría causar el abandono de la misión para cualquier piloto convencional. Por ejemplo, pueden volar cerca a bosques en

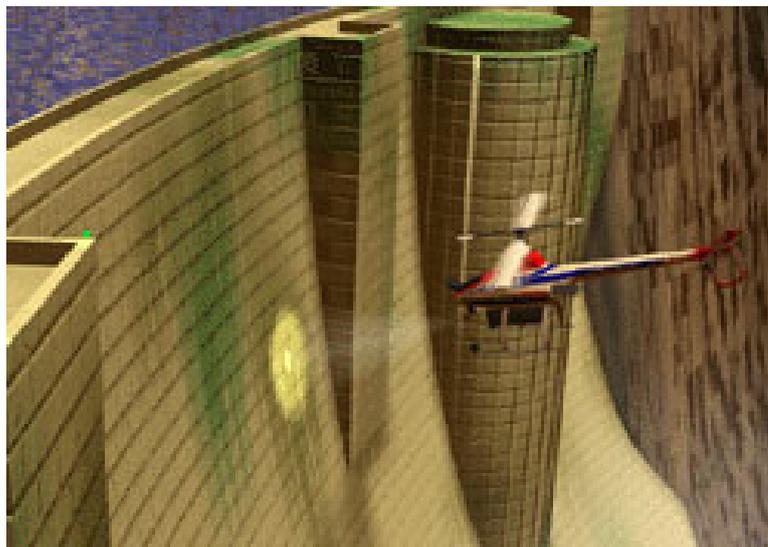


Figura 1.39: Mini helicóptero realizando inspección en una presa.

llamas buscando algún individuo, buscando áreas contaminadas, e identificando potenciales zonas radioactivas después de un accidente nuclear.



Figura 1.40: Mini helicóptero realizando tarea de búsqueda.



Figura 1.41: Mini helicóptero realizando patrullaje (izquierda). Mini helicóptero realizando persecución policíaca (derecha).

#### 1.3.4.4. Patrullaje y seguridad

Los helicópteros autónomos o tripulados remotamente también pueden ser usados para el patrullaje (ver figura 1.41 (izquierda)) en determinadas áreas y reportar actividades inusuales. Pueden permanecer patrullando áreas sin parar, y recargarse automáticamente de estaciones cercanas para continuar con la vigilancia. Los dispositivos incorporados a bordo pueden permitir volar detrás de vehículos policiales (ver figura 1.41 (derecha)) para ayudar en persecuciones de alta velocidad y también en operaciones de búsqueda criminal. Además, estacionados en los edificios más altos pueden enviar fotografías instantáneas a los expertos en la base control de lo que sucede abajo o de cualquier suceso importante.

#### 1.3.4.5. Mapeo aéreo

En una de las aplicaciones los helicópteros autónomos pueden ser usados para el mapeo aéreo (ver figura 1.42), construyendo mapas topológicos más precisos que los hechos por aviones convencionales.

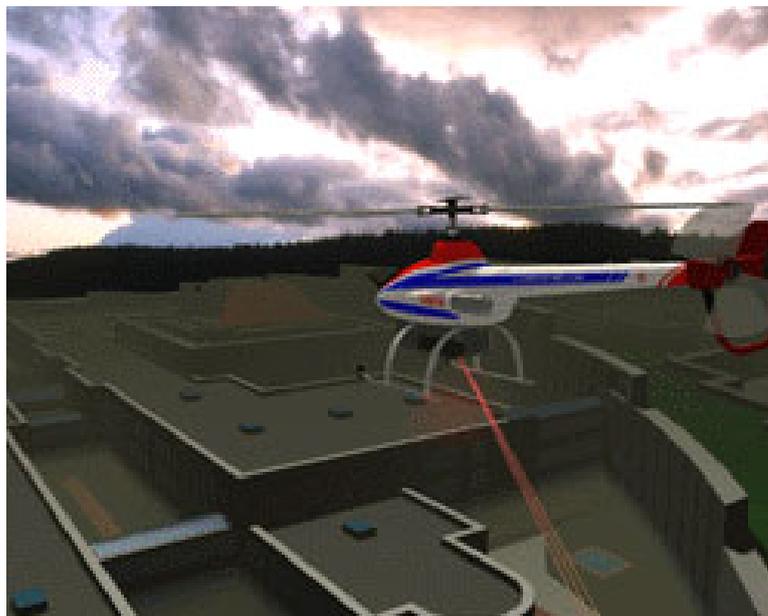


Figura 1.42: Mini helicóptero realizando mapeo aéreo.

#### 1.3.4.6. Cinematografía

Los mini helicópteros basados en visión también pueden operar cámaras filmadoras manteniendo el lente fijo en una escena, siguiendo objetos con su mecanismo de visión montado (ver figura 1.43). También pueden volar bajo trayectorias definidas haciendo posible la implementación de efectos especiales.

#### 1.3.4.7. Detección de banco de peces y fumigación

El instituto tecnológico de Tokio (Japón) ha desarrollado aplicaciones de búsqueda y localización de banco de peces, instalando helicóptero de cercano alcance para la ubicación de cardúmenes de atunes. Este mismo instituto ha desarrollado helicóptero con la capacidad de fumigar sembradíos de trigo (ver figura 1.44). Las ventajas principales son que el helicóptero puede volar a una altura menor y a una velocidad lenta, de tal manera que es aprovecha en mayor medida el producto fumigante [6, p. 14].



Figura 1.43: Mini helicóptero realizando tomas aéreas para una película.



Figura 1.44: Mini helicóptero realizando tareas de fumigación.

## 1.4. Teoría general del helicóptero

Antes de comenzar con la teoría general, es importante enunciar las partes más importantes del helicóptero. La figura 1.45 muestra las fuerzas y momentos resultantes que se originan como consecuencia de la interacción de las fuerzas en el rotor principal, fuselaje, cola, rotor de la cola y sistema de control. Las

ecuaciones que gobiernan estas interacciones están descritas por leyes físicas [6, p. 5].

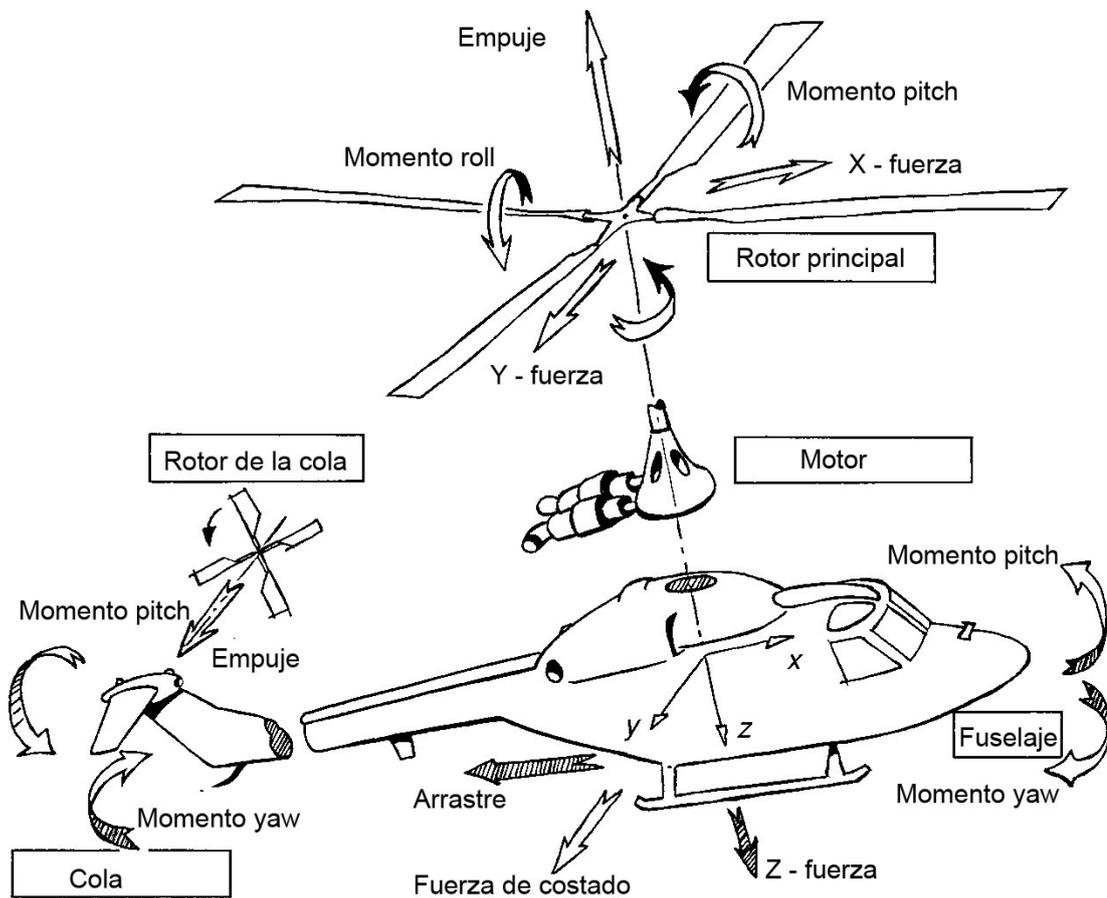


Figura 1.45: Componentes de para el modelado de un helicóptero

Varios modelos para helicópteros se han desarrollado, como es el caso a los modelos desarrollados por La Civita, Mettler, Padfield y Gavrillets. En los modelos iniciales el rotor principal fue considerado sólo como una fuerza vertical. Los movimientos de roll y pitch fueron acondicionados por alerones en los finales de las alas y por elevadores convencionales. Otro modelo consideró en cambio del pitch en la aspas como una función de la posición de las aspas [8, p. 9].

La configuración mas difundida del helicóptero es la que se muestra en la figura 1.46. El rotor principal proporciona el empuje para la sustentación y los controles para el roll y pitch del helicóptero. El rotor de la cola contrapesa el efecto del

torque producido por el rotor principal y además controla el yaw del helicóptero. La fuerza de sustentación es lograda variando el pitch o ángulo de ataque de las palas. El roll y pitch del fuselaje es logrado con ayuda del estabilizer bar [8, p. 10].

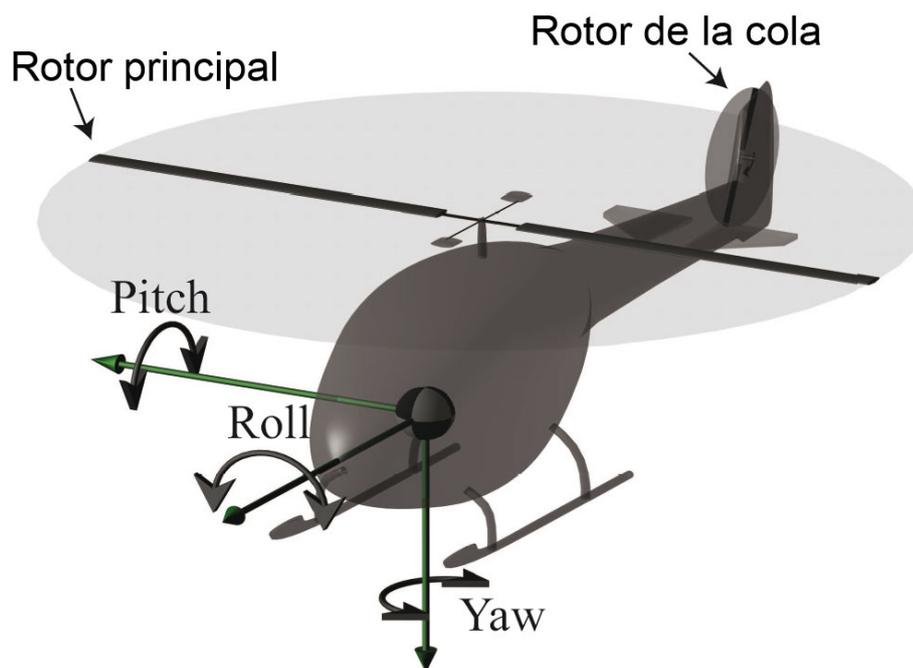


Figura 1.46: Ilustración del roll, pitch y yaw en el fuselaje del helicóptero.

### 1.4.1. Transmisión en el rotor principal

Cuatro son las entradas de control para un helicóptero, tres para el rotor principal y una para el rotor de la cola. Las entradas en el rotor principal son conectadas a través del “swash plate” como se aprecia en la figura 1.47. Como se puede observar en la figura el ángulo  $\theta_b$  es controlado por el transmisor del pitch que está conectado al swash plate.

Las entradas en el rotor principal están en dos categorías: colectivo y cíclico.

- El colectivo eleva totalmente el swashplate y aplica como resultado el mismo ángulo de ataque o ángulo pitch a ambas palas independientemente

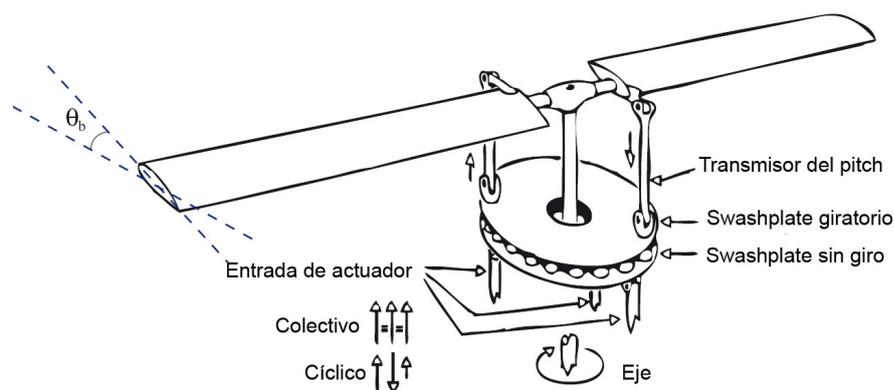


Figura 1.47: Se muestra la transmisión mecánica del swashplate con las palas del rotor principal, el cual por este medio controla el pitch de las palas.

de la posición en que se encuentren.

- El cíclico inclina el swash plate, el cual aplica una alteración del ángulo pitch de acuerdo a la posición de la pala. Esto implica ángulos pitch con diferente signos en cada pala.

Por tanto el colectivo se puede considerar como el controlador de la magnitud del empuje generado en el rotor principal, mientras que el cíclico gobierna la dirección del empuje. La entrada en el rotor de la cola es una entrada colectiva, como resultado se controla el empuje en la cola, el cual a su vez controla el yaw del helicóptero.

Algunos helicópteros, especialmente los de pequeña escala, están equipados con una barra estabilizadora. El cual afecta la entrada a través del sistema Bell-Hiller. De esta manera, el ángulo pitch de las palas serán controlados parcialmente por swash plate y parcialmente por la stabilizer bar, como aparece mostrado en la figura 1.48. Esta barra estabilizadora también permite disminuir la dinámica de movimiento cuando hay alteraciones en las entradas de control. También contribuye al pitch de las palas ya que está conectado a las palas de rotor principal [8, p. 11].

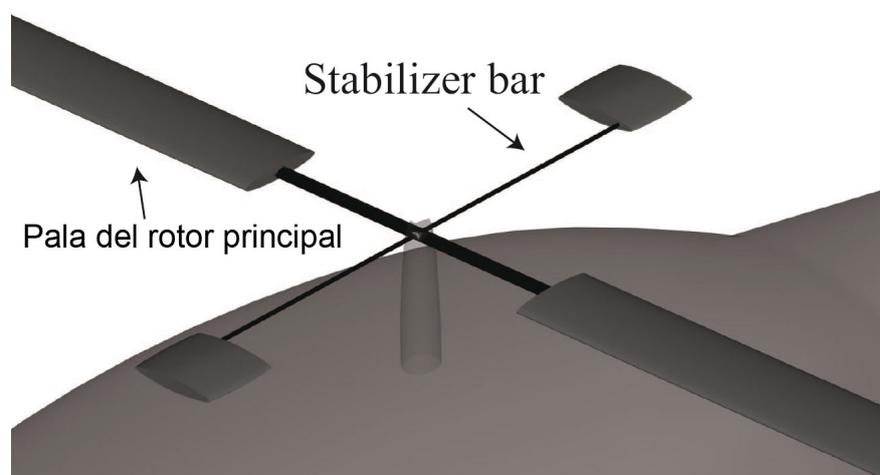


Figura 1.48: Se muestra la barra estabilizadora, la cual proporciona estabilidad al vuelo de los mini helicópteros.

### 1.4.2. Flapping o aleteo

Cuando la pala es angulada, esta produce un empuje lo cual hace que ésta se curve. Este fenómeno es conocido como flapping o aleteo, y es importante para el entendimiento de como los helicópteros son controlados. Un ejemplo del flapping se muestra en la figura 1.49. Los puntos extremos de las palas forman un plano conocido como Tip Path Plane (TPP). El Hub Plane está definido perpendicular al eje del rotor y por consecuencia es igual al TPP cuando no hay flapping. El TPP también es usado para definir el vector empuje generado por el rotor principal. Este vector puede ser definido actuando en el centro y perpendicular al TPP [8, p. 11].

Con el objetivo de describir como las entradas de control se propagan o transmiten por el sistema, vamos a describir el movimiento del Pitch.

### 1.4.3. Movimiento Pitch

Vamos a detallar el efecto, basado en la figura 1.50 y 1.51, de mover la barra cíclico hacia adelante. Esta es definida como una entrada longitudinal, donde las

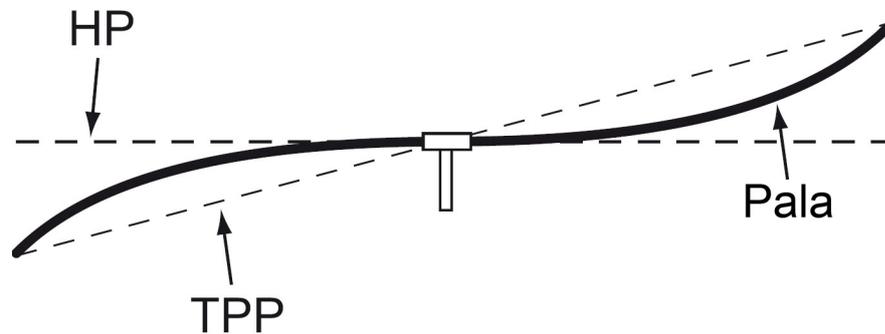


Figura 1.49: Los puntos extremos de las palas forman un plano conocido como TPP. El HP es igual al TPP cuando no hay flapping.

entradas de costado son conocidas como entradas laterales.

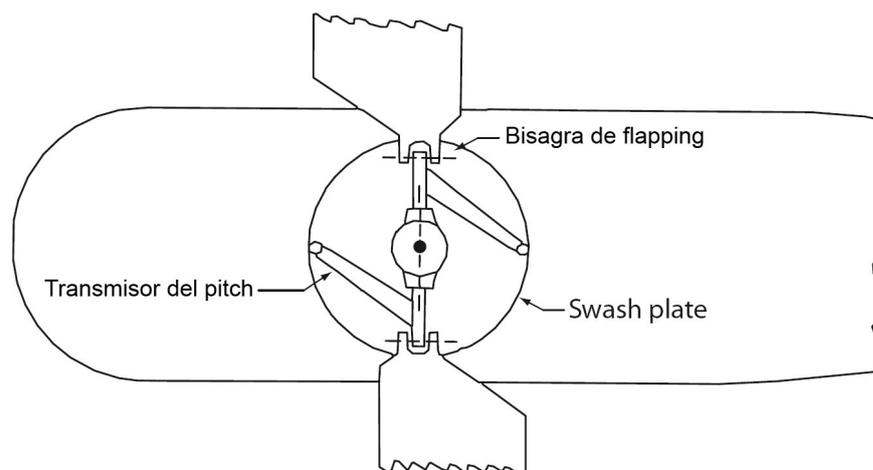


Figura 1.50: Vista superior del swash plate y la conexión a las palas.

Como se puede apreciar en la figura 1.51, una entrada longitudinal hacia adelante produce una inclinación del swash plate hacia adelante también. Debido al mecanismo de transmisión del swash plate y las palas del rotor, el TPP se inclina quedando posicionado paralelamente al swash plate. Por lo tanto para una entrada longitudinal del helicóptero experimenta un movimiento pitch (inclinación angular del fuselaje en dirección pitch). De la misma forma, si se aplica una entrada cíclica el helicóptero experimenta un movimiento roll

(inclinación angular del fuselaje en la dirección roll) [8, p. 12].

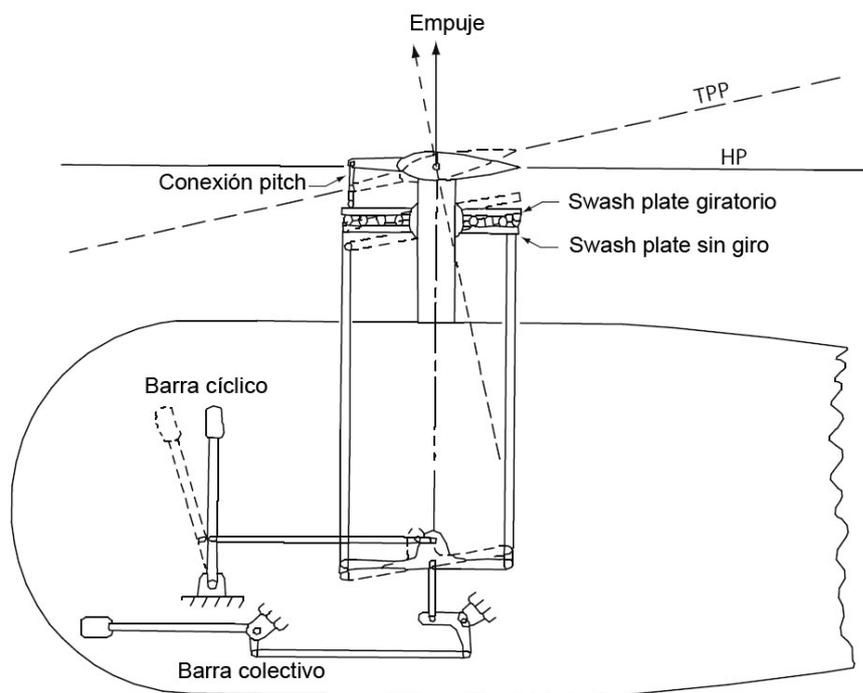


Figura 1.51: Vista de perfil de swash plate, sus conexiones y transmisiones.

## 1.5. Objetivos

1. Recopilar información especializada que permita tener un marco teórico completo y didáctico de la dinámica y aerodinámica de vuelo de un mini-helicóptero.
2. Desarrollar un modelo matemático no lineal que incluya las ecuaciones de movimiento más importantes que permitan simular la dinámica y aerodinámica de vuelo de un mini-helicóptero.
3. Elaborar e implementar un simulador de vuelo tridimensional que permita visualizar la respuesta dinámica de un mini-helicóptero sometido tanto a fuerzas internas (e.g. fuerza de sustentación) como fuerzas externas (e.g. fuerzas de arrastre o gravedad).

4. Desarrollar un algoritmo de estimación de posición a partir de la información de un sensor de barrido láser.
5. Implementar un controlador de posición bidimensional para el vuelo de un mini-helicóptero.

## 1.6. Alcances

1. En la parte del simulador se dará énfasis en la dinámica y aerodinámica de vuelo no lineal del mini-vehículo aéreo. El sistema funcionará en lazo abierto y por medio de un dispositivo de entrada como el teclado se podrá comandar sus movimientos. Mediante esta plataforma versátil se podrán simular diversos vehículos de vuelo de rotores giratorios como el helicóptero o el quadrotor. Además para el desarrollo del simulador se usarán datos y parámetros del helicóptero Caliber 5 de Kyosho.
2. Con respecto al controlador de vuelo del mini-helicóptero, se trabajará en el plano bidimensional; sin embargo, se estimarán los ángulos de Pitch, Roll, y Yaw (ángulos de Euler) para monitorear y restringir su posición angular. El controlador será implementado en la computadora a bordo del helicóptero quadrotor del Instituto de Robótica de la Universidad Carnegie Mellon. Cabe mencionar que este quadrotor tiene una mecánica totalmente acondicionada para el vuelo autónomo, un sistema de comunicación en tiempo real con la estación base en tierra, y un sistema de protección manual por si ocurriera un desperfecto.

# Capítulo 2

## Marco Teórico

### 2.1. Sistema de coordenadas

Para definir la posición del helicóptero en todo instante es importante definir un sistema de coordenadas fijo (Posición  $[0,0,0]$ ) y un sistema de coordenadas inercial y paralelo ubicados en el centro de gravedad del helicóptero el cual se asumirá como un cuerpo rígido. Estos sistemas de coordenadas van se van a relacionar entre si mediante los ángulos de Euler y las rotaciones de Euler.

#### 2.1.1. Angulos de Euler

Por lo general en sistemas 3D, usualmente necesitamos 3 rotaciones separadas respecto de nuestro sistema de coordenadas inercial a nuestro cuerpo rígido. Especialmente en vehículos aéreos. Existen muchas formas de hacer estas rotaciones (cambiando el orden de las rotaciones). Todas éstas son aceptables y algunas son más usadas que otras [14, p. 16]. La rotación estándar con sistema de coordenadas xyz alineado con el sistema inercial XYZ, realiza 3 rotaciones para reorientar las coordenadas de cuerpo rígido cite[p. 16]autor7. Dichas rotaciones estándar usadas en la dinámica del vuelo son yaw,  $\psi$ , pitch,  $\theta$ , y roll,  $\phi$  [15, p. 180], como se muestra en la figura 2.1.

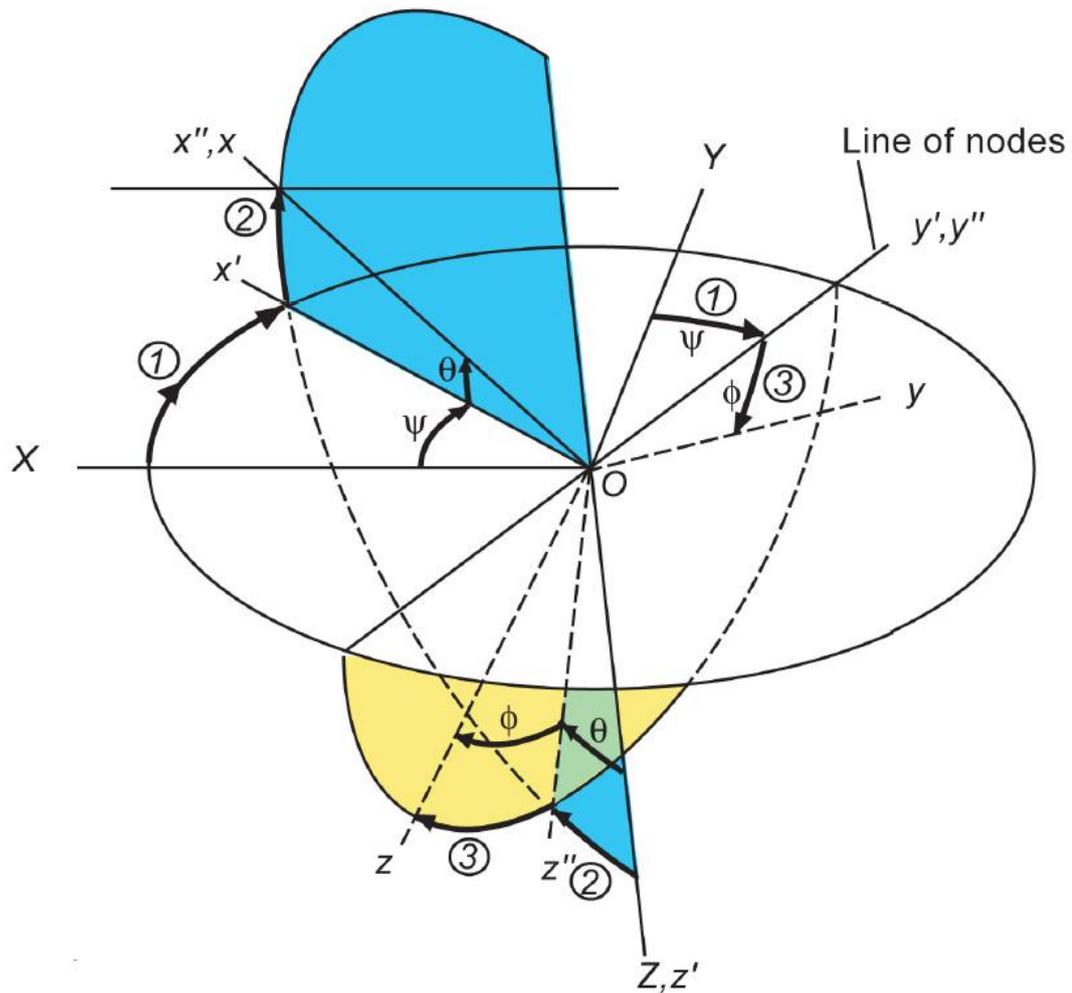


Figura 2.1: Angulos de Euler vistos de acuerdo a las rotaciones en los 3 ejes

Según la figura 2.1 y teniendo en cuenta la secuencia estándar, ésta se puede esquematizar:

- $\psi$  alrededor de  $z \rightarrow x'y'z'$ ,
- $\theta$  alrededor de  $y' \rightarrow x''y''z''$ ,
- $\phi$  alrededor de  $x'' \rightarrow XYZ$ .

Luego podemos escribir la matriz  $R_v$  (usando los términos  $c$  para el coseno y  $s$  para el seno) de transformación que relaciona el sistema de coordenadas inicial

y el sistema de coordenadas obtenido luego de hacer los 3 giros estándar:

$$R_v = \text{rot}(\psi, z)\text{rot}(\theta, y)\text{rot}(\phi, x). \quad (2.1)$$

Donde:

$$\text{rot}(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad (2.2)$$

$$\text{rot}(\theta, y) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad (2.3)$$

$$\text{rot}(\psi, z) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

$$R_v = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}. \quad (2.5)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_v \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (2.6)$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = R_v \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \quad (2.7)$$

En la ecuación (2.7) tenemos que la velocidad  $\mathbf{v}$  del sistema de coordenadas ubicado en el cuerpo rígido se puede expresar de la siguiente forma:

$$[\dot{x}, \dot{y}, \dot{z}]^T = [u, v, w]^T. \quad (2.8)$$

$$\mathbf{v} = [u, v, w]^T. \quad (2.9)$$

### 2.1.2. Rotaciones de Euler

Para conseguir la velocidad angular  $\omega$  en este caso necesitamos preocuparnos por estos tres términos [14, p. 18]:

- $\dot{\psi}$  alrededor de  $z$ ,
- $\dot{\theta}$  alrededor de  $y'$ ,
- $\dot{\phi}$  alrededor de  $x''$ .

Los cuales combinamos para obtener  $\omega$ . Como ya se había mencionado que las rotaciones estándar son: rotación en  $z$ , rotación en  $y$ , y finalmente rotación en  $x$ ; para la primera rotación  $\dot{\psi}$  alrededor de  $z = z'$ . En términos de  $X, Y, Z$  la dirección de rotación tiene componentes  $[0, 0, \dot{\psi}]^T$  igual que en  $x', y', z'$ , pero para transformar al vector de desde  $[x', y', z']^T$  hasta  $[x, y, z]^T$  deberíamos usar  $rot(\phi, x)rot(\theta, y)$ , el sentido es debido a que estamos haciendo la transformación desde un sistema de coordenadas anterior al sistema de referencia inercial. De la misma forma para  $\dot{\theta}$  alrededor de  $y' = y''$ . Solamente usamos  $rot(\phi, x)$  pues para llegar a  $y''$ , sólo necesitamos dicha rotación (ver figura 2.2). Por tanto usamos

$rot(\phi, x)$  para  $[0, \dot{\theta}, 0]^T$ . Finalmente para  $[\dot{\phi}, 0, 0]^T$  no habría problemas pues ésta coincide con su sistema de coordenadas inercial  $x, y, z$ .

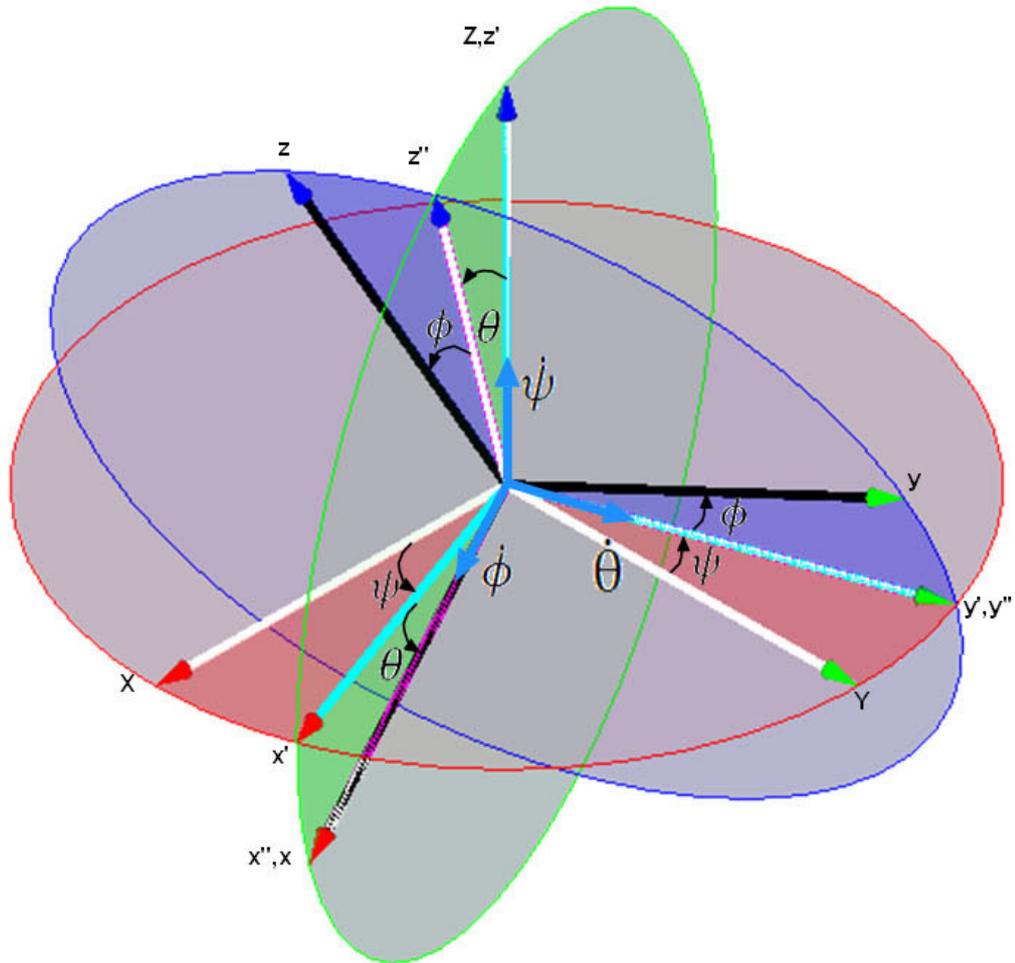


Figura 2.2: Rotaciones de Euler de acuerdo a la secuencia estándar.

Luego podemos escribir  $\omega = \vec{\dot{\phi}} + \vec{\dot{\theta}} + \vec{\dot{\psi}}$ ; sin embargo, estos vectores no forman entre sí un triedro mutuamente ortogonal. Por tanto, formamos las proyecciones ortogonales sobre las coordenadas  $x, y, z$  o simplemente  $xyz$ :

$$\omega = rot(\phi, x)^T rot(\theta, y)^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + rot(\phi, x)^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}. \quad (2.10)$$

Resultando:

$$R_\omega = \begin{bmatrix} \dot{\phi} - \sin(\theta) \dot{\psi} \\ \cos(\theta) \dot{\psi} \sin(\phi) + \dot{\theta} \cos(\phi) \\ \cos(\theta) \dot{\psi} \cos(\phi) - \dot{\theta} \sin(\phi) \end{bmatrix}. \quad (2.11)$$

O también:

$$\omega = R_\omega \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (2.12)$$

De la ecuación (2.11) tenemos la relación  $\omega$  respecto de los ángulos de Euler; no obstante, para las simulaciones la relación inversa que me muestra en la ecuación (2.13) es la más usada, aunque esta representación muestra singularidad en  $90^\circ$ .

$$R_\omega^{-1} = \begin{bmatrix} 1 & \frac{\sin(\theta) \sin(\phi)}{\cos(\theta)} & \frac{\sin(\theta) \cos(\phi)}{\cos(\theta)} \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix}. \quad (2.13)$$

## 2.2. Parámetros del mini helicóptero Caliber 5

Los parámetros físicos se muestran en la tabla 2.1. Los momentos de inercia fueron calculados alrededor de todo el fuselaje usando técnicas de péndulo torsional. En esta parte se han omitido los productos debido a la dificultad en su cálculo por medio de pruebas físicas.

Notar que m.r. se refiere al rotor principal y t.r. se refiere al rotor de la cola. Además es importante mencionar que algunos parámetros se consideraron con el mismo valor de la tabla de referencia de cite[p. 34]autor1. Esto se hizo debido a las similitudes geométricas existentes entre el Caliber 5 y el X-cell .60 del MIT. Por otro lado algunos nombres se han tomado como son conocidos en inglés debido a

que si se traducen podrían no tener una interpretación adecuada; adicionalmente, es más encontrarlos en la web con los nombres en inglés, así el lector será capaz de profundizar con mayor precisión.

Tabla 2.1: Parámetros del mini helicóptero Caliber 5

Parámetro	Descripción
$m = 3.4 \text{ kg}$	masa del helicóptero
$I_{xx} = 0.18 \text{ kg m}^2$	momento de inercia de rolling
$I_{yy} = 0.34 \text{ kg m}^2$	momento de inercia de pitching
$I_{zz} = 0.18 \text{ kg m}^2$	momento de inercia de yawing
$K_\beta = 54 \text{ N}\cdot\text{m}/\text{rad}$	rigidez torsional del hub)
$\gamma_{fb} = 0.8$	número lock de la barra estabilizadora
$B_{\delta_{lat}}^{nom} = 4.2 \text{ rad}/\text{rad}$	lateral cyclic to flap gain at nominal rpm
$A_{\delta_{lat}}^{nom} = 4.2 \text{ rad}/\text{rad}$	longitudinal cyclic to flap gain at nominal rpm
$K_\mu = 0.2$	escalamiento de respuesta de flap a la variación de velocidad
$\omega_{nom} = 167 \text{ rad}/\text{s}$	velocidad nominal del m.r.
$R_{mr} = 0.66 \text{ m}$	radio del m.r.
$c_{mr} = 0.058 \text{ m}$	cuerda del m.r.
$a_{mr} = 5.5 \text{ rad}^{-1}$	m.r. blade lift curve slope
$C_{D_0}^{mr} = 0.024$	m.r. blade zero lift drag coefficient
$C_{T_{max}}^{mr} = 0.0055$	máximo coeficiente de empuje del m.r.
$I_{\beta_{m.r.}} = 0.038 \text{ kg m}^2$	Inercia de blade flapping del m.r.
$R_{tr} = 0.13 \text{ m}$	radio del t.r.
$c_{tr} = 0.029 \text{ m}$	cuerda del t.r.
$a_{tr} = 5.0 \text{ rad}^{-1}$	t.r. blade lift curve slope
$C_{D_0}^{tr} = 0.024$	t.r. blade zero lift drag coefficient
Continuado en la siguiente página...	

Tabla 2.1 – Continuada	
Parámetro	Descripción
$C_{T_{max}}^{tr} = 0.05$	máximo coeficiente de empuje del t.r.
$n_{tr} = 4.66$	índice de transmisión t.r. a m.r.
$n_{es} = 9.0$	índice de transmisión ejde del motor a m.r.
$\delta_r^{trim} = 0.1 \text{ rad}$	t.r. pitch trim offset
$S_{vf} = 0.012 \text{ m}^2$	área efectiva del vertical fin
$C_{L_\alpha}^{vf} = 2.0 \text{ rad}^{-1}$	vertical fin lift curve slope
$\epsilon_{vf}^{tr} = 0.2 \text{ rad}^{-1}$	fraction of vertical fin area exposed to t.r. induced velocity
$S_{ht} = 0.01 \text{ m}^2$	área del horizontal fin
$C_{L_\alpha}^{ht} = 3.0 \text{ rad}^{-1}$	horizontal tail lift curve slope
$P_{eng}^{idle} = 0.0 \text{ W}$	idle engine power
$P_{eng}^{max} = 2000.0 \text{ W}$	idle engine power
$K_p = 0.01 \text{ s/rad}$	proportional governor gain
$K_i = 0.02 \text{ 1/rad}$	integral governor gain
$f_p^s = 12.5 \text{ Hz}$	rolling resonance frequency of the suspension system
$f_q^s = 9.0 \text{ Hz}$	pitching resonance frequency of the suspension system
$f_r^s = 9.6 \text{ Hz}$	yawing resonance frequency of the suspension system
$\xi^\sigma = 0.05$	damping ratio of the suspension system material
$S_x^{fus} = 0.1 \text{ m}^2$	frontal fuselage drag area
$S_y^{fus} = 0.22 \text{ m}^2$	side fuselage drag area
$S_z^{fus} = 0.15 \text{ m}^2$	vertical fuselage drag area
$h_{mr} = 0.235 \text{ m}$	m.r. hub height above c.g.
$l_{tr} = 0.91 \text{ m}$	t.r. hub location behind c.g.
$h_{tr} = 0.08 \text{ m}$	t.r. height above c.g.
$h_{ht} = 0.71 \text{ m}$	stabilizer location behind c.g.
Fin de Tabla	

Finalmente, es posible resumir los parámetros más importantes del mini-helicóptero como en la tabla 2.1; naturalmente, estos datos son aproximados y deben de ser reajustados al momento de la implementación o en el proceso de identificación o validación.

## 2.3. Ecuaciones de cuerpo rígido

Las ecuaciones de cuerpo rígido han sido ampliamente usadas para planeadores y aviones. Los modelos lineales son desprendidos de la dinámica de cuerpo rígido, esto permite analizar la estabilidad y control de los mencionados sistemas. En seguida vamos a describir el desarrollo de las ecuaciones de cuerpo rígido.

Es importante mencionar que el Helicóptero es un vehículo que simultáneamente se traslada y gira en 6 grados de libertad (6 GDL). La dinámica de cuerpo rígido de cuerpo esta descrita por las ecuaciones de movimiento de Newton-Euler. En la figura 4 podemos apreciar las variables del sistema de referencia inercial o solidario al cuerpo rígido con origen en el centro de gravedad del helicóptero.

### 2.3.1. Ecuaciones de Newton-Euler

Las ecuaciones de Newton-Euler expresan en el sistema de referencia inercial la conservación del momento lineal y angular. Para un vehículo de masa constante  $m$  y momento de inercia o tensor de inercia  $\mathbf{I}$ , las ecuaciones son:

$$m \frac{d^I}{dt} v = F \quad (2.14)$$

y

$$\mathbf{I} \frac{d^I}{dt} \omega = M. \quad (2.15)$$

Donde  $F = [XYZ]^T$  es el vector externo de fuerzas actuando sobre el centro de gravedad de vehículo y  $M = [LMN]^T$  es el vector de momentos externos. En el helicóptero las fuerzas externas y momentos externos son producidos por: el rotor principal y el rotor de la cola; las fuerzas gravitacionales, y fuerzas aerodinámicas producidas por el fuselaje y las superficies de la cola. La matriz o tensor de inercia del fuselaje viene dado por:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (2.16)$$

Donde  $I_{xx}, I_{yy}, I_{zz}$  los momentos de Inercia del fuselaje respecto de su centro de masa. Debido a que los productos de inercia son difíciles de medir y requieren de un equipo de balanceo; estos fueron despreciados como se muestra en (2.15) [5, p. 32]. Las ecuaciones dinámicas para el movimiento del helicóptero son sustantivamente simplificadas al expresarlas en las coordenadas de cuerpo rígido. En éstas condiciones, la relación de cuerpo rígido de Euler es [13, p. 55]:

$$\begin{bmatrix} mI_{den_{3 \times 3}} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mv \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ M \end{bmatrix}. \quad (2.17)$$

En fuerzas:

$$m \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} wq - vr \\ ur - wp \\ vp - uq \end{bmatrix} = F. \quad (2.18)$$

En momentos:

$$\begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} = \begin{bmatrix} qr (I_{yy} - I_{zz}) \\ pr (I_{zz} - I_{xx}) \\ pq (I_{xx} - I_{yy}) \end{bmatrix} + \begin{bmatrix} L \\ M \\ N \end{bmatrix}. \quad (2.19)$$

La fuerza de gravedad actuando en el helicóptero respecto al sistema inercial es:

$$\mathbf{G} = R_v^{-1} \mathbf{g} = \begin{bmatrix} -g \sin(\theta) \\ g \sin(\phi) \cos(\theta) \\ g \cos(\phi) \cos(\theta) \end{bmatrix}. \quad (2.20)$$

Donde  $\mathbf{g} = [0, 0, g]^T$ .

## 2.4. Modelo aerodinámico del mini helicóptero

Existe una extensa literatura relacionada a la dinámica de helicópteros de escala completa. Modelos desarrollados paso a paso se han planeado y publicado [4,7]. Los modelos usados en helicópteros a escala completa, usados en simuladores actuales, son típicamente de un alto orden y contienen una gran cantidad de parámetros que no se pueden medir directamente. Seguidamente estos modelos requieren de un proceso de validación hasta que dichos modelos puedan predecir el comportamiento dinámico de manera precisa. Aplicar estas técnicas de Modelamiento podría resultar en una tarea complicada dada las diferencias que existen entre helicópteros en miniatura y helicópteros de escala completa. Los helicópteros presentan una gran cantidad de condiciones aerodinámicas, donde complejas interacciones se presentan entre el rotor principal y el fuselaje o la cola. En mini-helicópteros, estos efectos no obvios tienden a ser magnificados por las grandes fuerzas y momentos en el rotor que se generan casi instantáneamente las entradas de control en el rotor. Los modelos lineales, inicialmente encontrados

utilizando técnicas de identificación, has sido inadecuados para describir el comportamiento de los mini-helicópteros a bajas frecuencias. Debido a que los modelos a escala pequeña tienen grandes desviaciones en las variables de estado (actitud, velocidad angular, y velocidad) [5, p. 31].

Mettler [3], aplicó un método de identificación de sistemas en el dominio de la frecuencia para estudiar las características de mini-helicópteros. Él desarrolló e identificó modelos parametrizados linealizados para el vuelo suspendido y cruzado en el Yamaha R-50. Más tarde, él aplicó el mismo modelo parametrizado para el X-Cell .60, validando y concluyendo que las fuerzas y momentos en el rotor dominan preponderantemente la respuesta dinámica de los mini-helicópteros. Esto simplifica enormemente la tarea del modelado. Las condiciones de vuelo son modeladas mediante la teoría de un cuerpo rígido aumentado a una dinámica de rotor y estabilizador de primer orden; las dinámicas de influjo no fueron necesarias. Las ecuaciones de rotor y estabilizador pueden ser asignadas en un ecuación de movimiento del rotor efectivo de primer orden. Los modelos lineales capturan la dinámica de vehículo con precisión, y esto es válido en una amplia región alrededor del punto de operación. El modelo predice de manera precisa la respuesta de velocidad angular para entradas de maniobra violentas [5, p. 32].

Es importante mencionar algunas características preliminares del mini-helicóptero [22, p. 27]:

- Los mini-helicópteros son más ágiles comparados a los modelos de escala completa.
- Es posible realizar maniobras controladas completamente por computador.
- Pueden producirse empujes negativos, lo cual permite un vuelo invertido.
- Es posible realizar maniobras extremas

- Los momentos de inercia decrecen con una quinta potencia del factor de escala.
- El empuje del rotor decrece con una tercera potencia del factor de escala.
- Tiene una alta razón empuje/peso (2/3) lo que acentúa los efectos de escalamiento.

En resumen, en esta parte del capítulo vamos a dar a conocer la lista de parámetros con sus valores numéricos, las ecuaciones dinámicas de movimiento, y las expresiones de fuerzas y momentos aplicados al helicóptero.

### 2.4.1. Ecuaciones de movimiento

Las ecuaciones de movimiento se desprende de las ecuaciones de Newton-Euler descrito anteriormente (Los productos de inercia han sido despreciados) [5, p. 33].

$$\dot{u} = vr - wq - g \sin(\theta) + \frac{X_{mr} + X_{fus}}{m}, \quad (2.21)$$

$$\dot{v} = wp - ur + g \sin(\phi) \cos(\theta) + \frac{Y_{mr} + Y_{fus} + Y_{tr} + Y_{vf}}{m}, \quad (2.22)$$

$$\dot{w} = uq - vp + g \cos(\phi) \cos(\theta) + \frac{Z_{mr} + Z_{fus} + Z_{ht}}{m}, \quad (2.23)$$

$$\dot{p} = \frac{qr(I_{yy} - I_{zz})}{I_{xx}} + \frac{L_{mr} + L_{vf} + L_{tr}}{I_{xx}}, \quad (2.24)$$

$$\dot{q} = \frac{pr(I_{zz} - I_{xx})}{I_{yy}} + \frac{M_{mr} + M_{ht}}{I_{yy}}, \quad (2.25)$$

$$\dot{r} = \frac{pq(I_{xx} - I_{yy})}{I_{zz}} + \frac{-Q_e + N_{vf} + N_{tr}}{I_{zz}}. \quad (2.26)$$

Estas fuerzas y momentos actuantes en el helicóptero están organizadas por sus componentes:  $()_{mr}$  para el rotor principal;  $()_{tr}$  para el rotor de la cola;  $()_{fus}$  para el fuselaje (incluyendo efectos aerodinámicos);  $()_{vf}$  para el fin vertical y  $()_{ht}$  para el estabilizador horizontal. Estas fuerzas y momentos están mostrados en la figura 2.3.

$Q_e$  es el torque producido por el motor para contrarrestar el torque aerodinámico producido en las palas del rotor principal. Las palas giran en el sentido de la agujas del reloj vistas desde arriba; por tanto,  $0 \leq Q_e$ . En la ecuaciones de arriba asumimos que el centro de gravedad (c.g.) coincide con el centro de presiones; por eso, los momentos creados por las fuerzas aerodinámicas en el fuselaje fueron despreciados [5, p. 33]. Las velocidades son obtenidas a partir de las velocidades del sistema solidario al cuerpo rígido por las matrices de transformación descritas anteriormente, éstas son integradas para obtener las posiciones. El método de integración fue Euler con un tiempo de muestreo de 0.001 segundos.

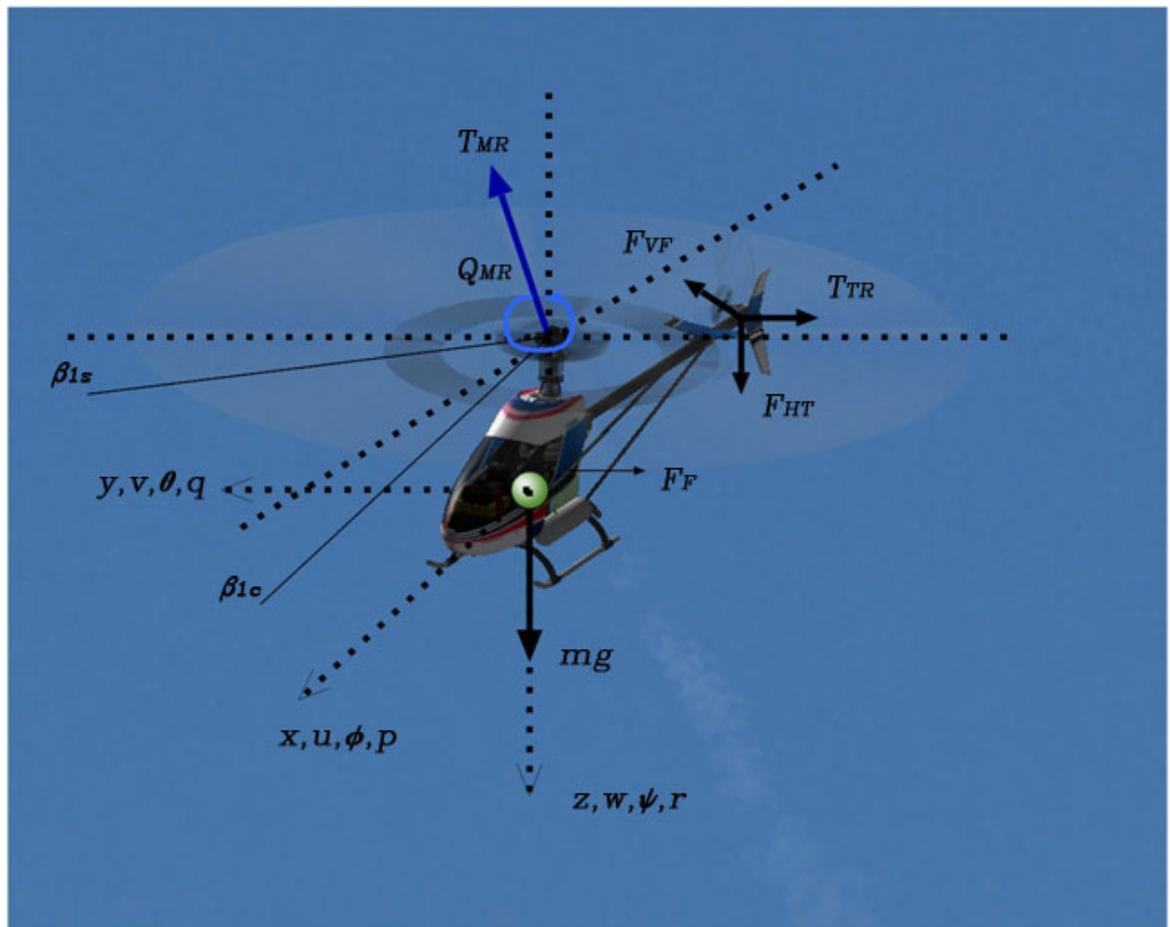


Figura 2.3: Momentos y Fuerzas actuando en el Helicóptero

### 2.4.2. Influjos (*Inflow*) del rotor

El influjos del rotor es el nombre dado al flujo de campo inducido por el rotor en el disco del rotor, de esa manera contribuye a la incidencia de pala local y a la presión dinámica [15, p. 115]. Asumiremos que para el análisis de la dinámica de vuelo será suficiente considerar la componente normal del influjos. La forma más fácil de representar el rotor wake (o vórtices generados en los contornos de las palas) está basada en la teoría del disco actuador (actuator disk), un artificio matemático representa al rotor como un número infinito de palas, capaces de generar aire a través del disco para soportar el salto de presión a través de este. De la figura 2.4, notamos que  $T$  es el empuje del rotor,  $v$  es la velocidad a varios estados del flujo,  $v_i$  el influjos en el disco;  $V_c$  la velocidad de ascenso y  $V_d$  es la velocidad de descenso del rotor [15, p. 116].

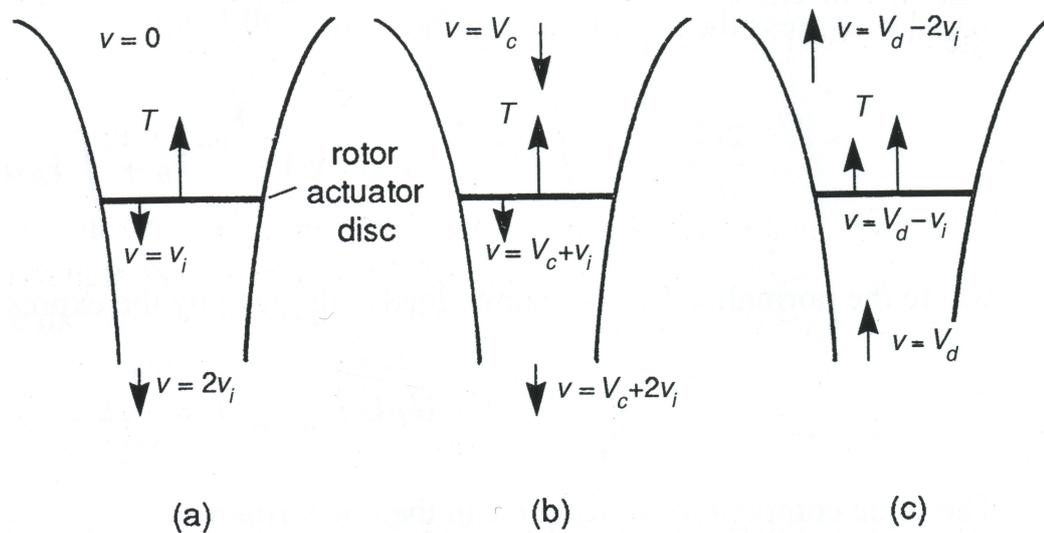


Figura 2.4: Estados del flujo en el rotor en movimiento axial: (a) suspensión (hover); (b) ascenso (climb); (c) descenso (descent)

### 2.4.3. Fuerzas y momentos en el rotor principal

#### 2.4.3.1. Empuje

Para el rotor principal asumimos que el influjo esta en estado estable y uniforme. Según [15, p. 127] la constante para la instalación del influjo transitorio en hover es dado por:

$$\tau_\lambda = \frac{0.849}{4\lambda_{trim}\Omega_{mr}}. \quad (2.27)$$

Para nuestro helicóptero la velocidad inducida en hover estable puede ser determinada facilmente de la teoría del momentum [15, p. 117]:

$$V_{imr} = \sqrt{\frac{mg}{2\rho\pi R_{mr}^2}} = 3.18 \frac{\text{m}}{\text{s}}. \quad (2.28)$$

La tip speed (velocidad del extremo de la pala) en el rotor principal es  $V_{mr}^{tip} = \Omega_{mr}R_{mr} = 110.22 \frac{\text{m}}{\text{s}}$ , por lo cual el indice de influjo es  $\lambda_{imr} = \frac{V_{imr}}{V_{mr}^{tip}} = 0.029$ . Por tanto el tiempo que le toma al influjo establecerse es de  $\tau_\lambda = 0.043 \text{ s}$ , el cual es significativamente más rápido que la dinámica de cuerpo rígido [5, p. 35]. Adicionalmente, como la autonomía del helicóptero es dominada por la rigidez torsional de las palas de rotor principal, el manejo del influjo transitorio es menos crítico [5, p. 35].

La teoría del momentum vista de manera iterativa proporciona un esquema apropiado para estimar el coeficiente de empuje y el índice de influjo como una función de la velocidad de aire, la velocidad del rotor y la entrada colectiva (collective) [15, p. 124]. Se van a ignorar los ángulos de flapping (aleteo) para el cálculo del empuje en el rotor. Las palas del rotor principal no tienen ninguna torcedura. La influencia de las entradas cíclicas (lateral y longitudinal) en el empuje son de segundo orden para nuestro rango de índice de avance (advance ratio)  $\mu < 0.15$ , pero fueron ignorados [5, p. 35]. Los coeficientes que se vayan a calcular de ahora en adelante tiene carácter empírico; el coeficiente para el empuje

en el rotor principal (m.r.) viene dado por:

$$C_{T_{mr}} = \frac{T_{mr}}{\rho (\Omega_{mr} R_{mr})^2 \pi R_{mr}^2}, \quad (2.29)$$

donde  $T$  es el empuje. Luego los siguientes sistemas de ecuaciones pueden ser resueltos iterativamente:

$$\lambda_0 = \frac{C_T}{2 \eta_\omega \sqrt{\mu^2 + (\lambda_0 - \mu_z)^2}}, \quad (2.30)$$

$$C_T^{ideal} = \frac{a\sigma}{2} \left( \theta_0 \left( \frac{1}{3} + \frac{\mu^2}{2} \right) + \frac{u_z - \lambda_0}{2} \right), \quad (2.31)$$

$$\begin{cases} C_T^{ideal} & -C_T^{max} \leq C_T^{ideal} \text{ y } C_T^{ideal} \leq C_T^{max} \\ -C_T^{max} & C_T^{ideal} < -C_T^{max} \\ C_T^{max} & C_T^{max} < C_T^{ideal} \end{cases}, \quad (2.32)$$

donde:

$$C_T^{max} = \frac{T^{max}}{\rho (\Omega R)^2 \pi R^2}, \quad (2.33)$$

- $\mu = \frac{\sqrt{(u-u_{wind})^2 + (v-v_{wind})^2}}{\Omega R}$  - índice de avance
- $u_z = \frac{w-w_{wind}}{\Omega R}$  - componente normal del flujo de aire
- $\sigma = \frac{2c}{\pi R}$  - pendiente de la curva de subida de la pala
- $a$  - índice de solidez o consistencia
- $\theta_0$  - ángulo colectivo de entrada
- $\eta_\omega$  - coeficiente no ideal de contracción de vórtice (wake contraction)
- $T^{max} = 2.5mg$  - máximo empuje del rotor principal

Introducimos  $\eta_\omega$  para tomar en cuenta la wake contraction no ideal y la pérdida de potencia por la velocidad no uniforme y la distribución de presión

en el vórtice. De acuerdo con [5, p. 36],  $\eta_\omega$  puede ser aproximado a 0.9 (Debido a las características similares en dimensiones con el X-Cell .60 vamos a asumir que para el Caliber 5, éste coeficiente es el mismo).

Debemos de notar que la simulación del sistema de mini helicóptero no representa adecuadamente la dinámica del helicóptero para las condiciones de vortex-ring (anillos que se forman a los extremos de la palas del rotor principal y de la cola, reduciendo la fuerza de sustentación como se ve en la figura 2.5).

La teoría del momentum mostrada previamente es adecuada para estimar el empuje en el rotor en condiciones de hover (suspensión) y forward (avance).

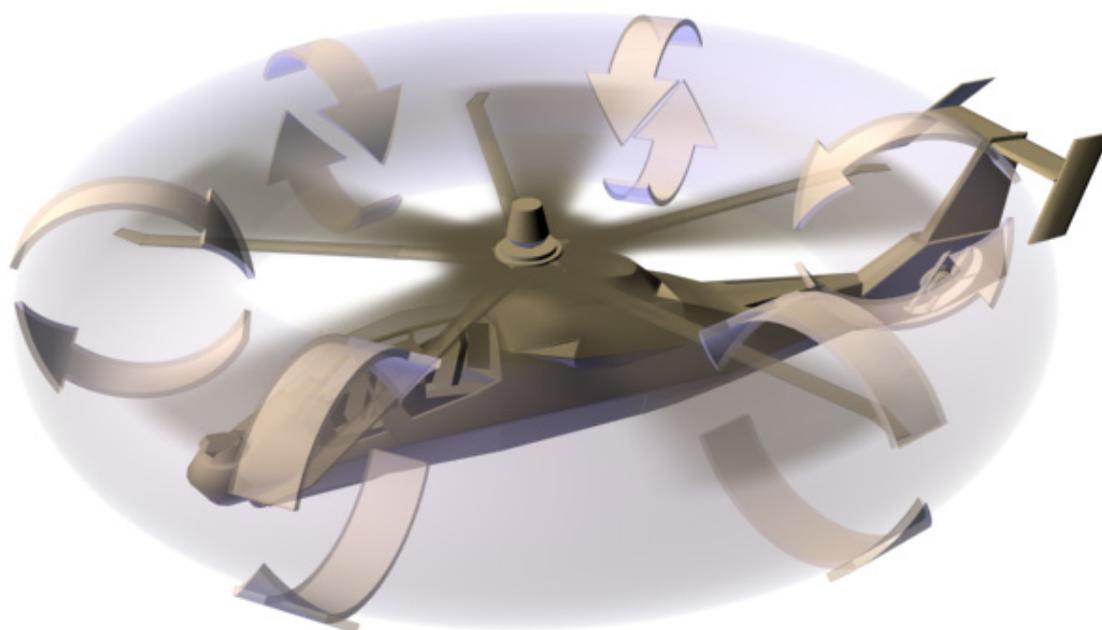


Figura 2.5: Las flechas curvadas indican el flujo de aire alrededor del disco del rotor. El helicóptero mostrado es el RAH-66 Comanche

#### 2.4.3.2. Torque

El torque en el rotor principal puede ser calculado a aproximadamente como la suma del torque inducido por el empuje generado , y el torque generado por el

perfil de arrastre de la palas [15, p. 115].

$$C_Q = \frac{Q}{\rho (\Omega R)^2 \pi R^3} = C_T (\lambda_0 - \mu_z) + \frac{C_{D0}\sigma}{8} \left(1 + \frac{7}{3}\mu^2\right), \quad (2.34)$$

donde  $C_Q$  es el coeficiente de torque,  $C_{D0}$  es el coeficiente de perfil de arrastre en la pala del rotor principal. El perfil de arrastre no es afectado significativamente por los cambios en el collective [5, p. 37]. Por tanto, el momento producido por el rotor principal está dado por:

$$Q_{mr} = C_Q \rho (\Omega R)^2 \pi R^3. \quad (2.35)$$

El coeficiente de arrastre del perfil del rotor principal es  $C_{D0} = 0.024$ . Una sub-estimación de este parámetro podría conducir a una sobre-predicción de la velocidad del rotor en condiciones de windmilling (en autorotación o maniobras violentas) [5, p. 38].

#### 2.4.3.3. Momentos en el rotor principal y dinámica del flapping

El angulo flapping *beta* del rotor principal puede ser representado por la serie de Fourier del ángulo  $\psi$ , considerando solamente los tres primeros coeficiente [15, p. 35],[13, p. 70]:

$$Q_{mr} = C_Q \rho (\Omega R)^2 \pi R^3. \quad (2.36)$$

El flapping de la barra estabilizadora puede ser representado por una ecuación similar sin considerar el término constante ya que no se forma ningún conoado (*coning*) [5, p. 38],[13, p. 80]:

$$\beta_s(\psi) = a_0 + a_s \sin(\psi) + b_s \cos(\psi). \quad (2.37)$$

La barra estabilizadora contribuye al cambio del pitch en la paleta del rotor principal a través de la unión mecánica, que se muestra en la siguiente ecuación:

$$\theta(\psi) = \theta_0 + \theta_{lon} \sin(\psi) + \theta_{lat} \cos(\psi) + k_s \beta_s. \quad (2.38)$$

Las deflexiones en el swashplate (plato transmisor de movimiento) cambian el cyclic pitch del rotor principal y de la barra estabilizadora (stabilizer bar). Acoplados por ecuaciones diferenciales de segundo orden los cuales pueden ser manejados por medio del uso de coeficientes de la serie de Fourier.

El número Lock es un coeficiente de escalamiento no-dimensional muy importante, el cual da la escala aerodinámica de similitud para las fuerzas de inercia que actúan en las palas de rotor. El número lock se define como [15, p. 34]:

$$\gamma = \frac{\rho c a R^4}{I_\beta}. \quad (2.39)$$

Para las palas del rotor principal el número Lock es relativamente grande,  $\gamma_{mr} \approx 3.7$  [5, p. 39]. Por tanto, el movimiento flapping (flapping motion) está bien amortiguado. Para una entrada escalón le corresponde un tiempo de asentamiento (con el 5% de error en estado estacionario) igual a  $24/(\gamma \Omega)$ . Para la barra estabilizadora, con pequeñas superficies aerodinámicas, el número Lock es pequeño  $\gamma_{fb} \approx 0.8$ , y con un correspondiente tiempo de establecimiento de 0.144.

El rotor principal y la barra estabilizadora pueden ser conectados por medio de la dinámica de Tip-Path-Plane (TPP) con dos estados. Este resultado basado en la identificación y comparación en el dominio de la frecuencia de las funciones de transferencias completas y reducidas para la dinámica de attitude (actitud) [5, p. 39].

#### 2.4.3.4. Dinámica en modo Tip-Path-Plane

Si contáramos la conicidad, la dinámica de rotor TTP tendría tres modos de movimiento oscilatorio: el modo conicidad, avance y retroceso [13, p. 71]. El modo conicidad es caracterizado por el blade flapping (inclinación del la paleta principal hacia arriba) que resulta cuando se cambia el ángulo  $\beta_0$  (ver figura 2.6). Su frecuencia natural es igual a frecuencia natural de blade flapping  $\lambda_\beta$ . Ese modo es cambiado típicamente por variaciones carga en el rotor. El avance corresponde al cabeceo o nutation-type del cono del rotor. Este modo tiene una frecuencia natural de orden  $\lambda_\beta + 1$  lo cual representa cerca de dos veces la velocidad angular del rotor [13, p. 71]. El modo retroceso o precession-type del cono del rotor, con una frecuencia natural igual a carga en el rotor. El avance corresponde al cabeceo del cono del rotor. Este modo tiene una frecuencia natural de orden  $\lambda_\beta - 1$ .

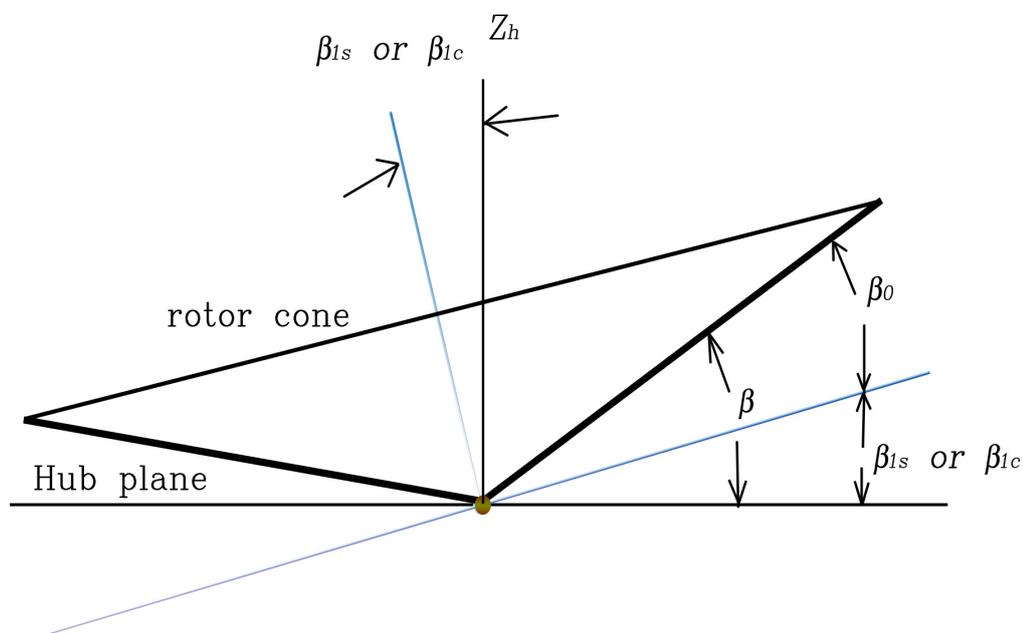


Figura 2.6: Representación del Tip-Path-Plane en el rotor. El ángulo  $\beta$  está representado por la serie de Fourier, con término no periódico  $\beta_0$  el cual describe la conicidad y los primeros armónicos  $\beta_{1s}$  (o  $\beta_{1c}$ ) describen la inclinación lateral (longitudinal) del ángulo del TPP.

### 2.4.3.5. Ecuaciones de primer orden de Tip-Path-Plane

Para la simulación de nuestro sistema solo las fuerzas que tienen implicancia en el fuselaje van a ser consideradas. Por tanto éste análisis se reduce a estudiar la dinámica de flapping. Nosotros representamos la dinámica de flapping lateral y longitudinal de la siguiente forma [5, p. 39]:

$$\dot{b}_1 = -p - \frac{b_1}{\tau_e} + \frac{1}{\tau_e} \left( \frac{\partial(b_1)}{\partial(\mu_v)} \frac{v - v_w}{\Omega R} \right) + \frac{B_{\delta_{lat}}}{\tau_e} \delta_{lat} \quad (2.40)$$

y

$$\dot{a}_1 = -q - \frac{a_1}{\tau_e} + \frac{1}{\tau_e} \left( \frac{\partial(a_1)}{\partial(\mu)} \frac{u - u_w}{\Omega R} + \frac{\partial(a_1)}{\partial(\mu_z)} \frac{w - w_w}{\Omega R} \right) + \frac{A_{\delta_{lon}}}{\tau_e} \delta_{lon}. \quad (2.41)$$

Donde  $B_{\delta_{lat}}$  y  $A_{\delta_{lon}}$  son las ganancias efectivas en estado estacionario para la entradas cíclicas en los ángulos del rotor.  $\delta_{lon}$  y  $\delta_{lat}$  son las entradas longitudinal y lateral (del radio control o las salidas del controlador).  $u_w$ ,  $v_w$  y  $w_w$  son las velocidades efectivas del viento en la direcciones  $X$ ,  $Y$ , y  $Z$  respectivamente;  $\tau_e$  es la constante de tiempo efectiva del rotor considerando la barra estabilizadora. De acuerdo con [15, p. 35] los coeficientes de acoplamiento de flapping para pitch y roll tienen un orden de magnitud menor que los coeficientes directos que se muestran en las ecuaciones (2.40) y (2.41); por lo tanto, al aplica una entrada pitch neta, el roll permanecerá muy cercano a cero y viceversa. Lo cual lo hace particularmente atractivo.

Los momentos dominantes del rotor son los momentos producidos por la deflexión de las paletas en el rotor principal. A continuación vamos a describir los momentos generados en la dirección roll (resultado del deflexión TPP lateral  $b_1$ ). La figura 2.7 muestra lo momentos del rotor actuando en el rotor principal. La primera componente es debido a la deformación angular (restricción física) de la pala unida al rotor principal. Esta restricción física puede ser aproximada usando

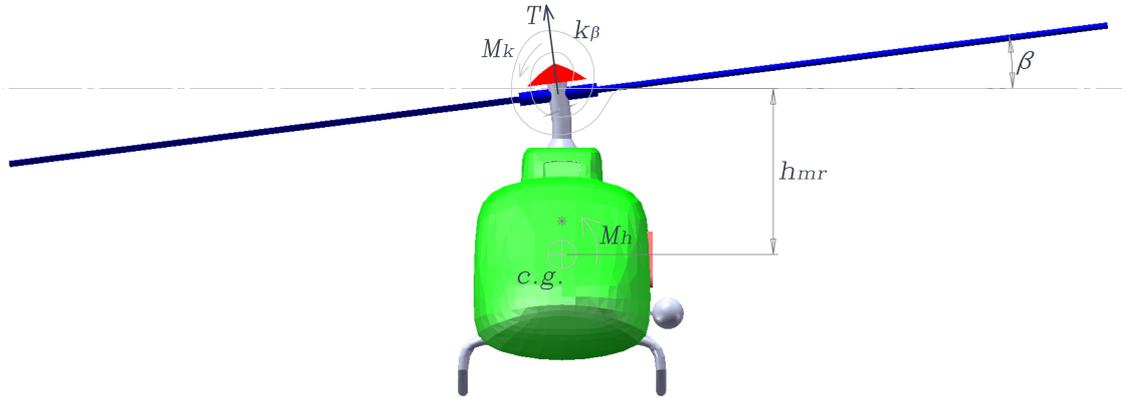


Figura 2.7: Momentos del rotor actuando sobre el fuselaje

un coeficiente de torsión (de la pala con respecto al mástil del rotor principal)  $K_\beta$ ; en consecuencia, esto genera un momento en roll  $M_{k,lat} = K_\beta b_1$ . La segunda componente resulta de la inclinación del vector de empuje (asumiendo que dicho vector es perpendicular al TPP). Este vector generará un momento proporcional a los ángulos de flapping o deflexión angular. El brazo es  $h_{mr}$  el cual es la distancia entre la cabeza del rotor y el centro del gravedad del helicóptero. Finalmente, todo esto resulta en un momento lateral  $M_{h,lat} = T h_{mr} b_1$ . Por tanto, el momento total del rolling en el centro del gravedad del helicóptero, el cual entraría en la ecuación de cuerpo rígido (2.24) es:

$$L_{mr} = (K_\beta + T h_{mr}) b_1. \quad (2.42)$$

Análogamente el momento pitching es:

$$M_{mr} = (K_\beta + T h_{mr}) a_1. \quad (2.43)$$

De acuerdo con [13, p. 114] la constante de tiempo para el sistema rotor-barras

estabilizadora tiene una valor aproximado de:

$$\tau_e = \frac{16}{\lambda_{fb} \Omega_{mr}} \approx 0.1 \text{ s.} \quad (2.44)$$

Donde  $\lambda_{fb}$  es el Lock number del flybar (barra estabilizadora) que se muestra en la ecuación (??). Además las ganancias en estado estable dependen del mecanismo de transmisión del swashplate. De acuerdo con [5, p. 41] estos valores son aproximados como una función efectiva de la presión dinámica:

$$B_{\delta_{lat}} = B_{\delta_{lat}}^{nom} \left( \frac{\Omega}{\Omega_{nom}} \right)^2, \quad (2.45)$$

$$A_{\delta_{lon}} = A_{\delta_{lon}}^{nom} \left( \frac{\Omega}{\Omega_{nom}} \right)^2. \quad (2.46)$$

Estas ganancias fueron determinadas mediante un emparejamiento de la velocidades angulares como resultado de entradas escalones en los cíclicos (longitudinal y lateral). Es importante recordar que el modelo cubre gran parte de la dinámica del helicóptero: desde el vuelo en hover hasta el vuelo en forward a 20 m/s, ésta máxima velocidad de avance corresponde a un índice de avance de  $\mu = 0.15$ , es cual es relativamente bajo, lo cual permite asumir, por ejemplo, que el empuje es perpendicular al disco del rotor.

El flapping (aleteo) debido a la velocidad translacional es decrito por las derivadas de flapping  $\frac{\partial(a_1)}{\partial(\mu)}$  y  $\frac{\partial(b_1)}{\partial(\mu_v)}$ . Un valor teórico para el flapping longitudinal de estado estable sin considerar la barra estabilizadora esta dado, según [4, p. 107], por:

$$a_1 = \frac{2\mu(4/3\delta_{col} - \lambda_0)}{(1 + 3/2\mu^2)} \approx 2\mu(4\delta_{col}/3 - \lambda_0). \quad (2.47)$$

Una expresión similar se encuentra en [15, p. 35]. De (2.47) la barra estabilizadora reduce le respuesta del sistema del rotor a turbulencias. Dado que esta derivada cumple un rol importante en la frecuencia y amortiguamiento en modo Phugoid, el

cual es muy lento; también, es difícil de calcular con el método de identificación en el dominio de la frecuencia [5, p. 42]. A continuación introducimos un coeficiente de escalamiento para (2.47) y lo linealizamos para encontrar [5, p. 42]:

$$\frac{\partial(a_1)}{\partial(\mu)} = 2 K_\mu(4/3 \delta_{col} - \lambda_0). \quad (2.48)$$

Una estimación rápida para el coeficiente  $K_\mu$  puede ser obtenida por medio de un emparejamiento o comparación con la entrada cíclica en estado estable en vuelo de avance a una velocidad constante con un valor predeterminado en simulación con características similares. Para el X-Cell 60 el valor de  $K_\mu = 0.2$ , dicho valor reduce la respuesta al flapping en estado estable en un factor de 5. Además de la ecuación (2.47) y de la simetría en el rotor, podemos concluir que las derivadas longitudinal y lateral son iguales en magnitud, y en ambos casos esto causa que el rotor se mueva alejándose del aire entrante [5, p. 43]:

$$\frac{\partial(b_1)}{\partial(\mu_v)} = -\frac{\partial(a_1)}{\partial(\mu)}. \quad (2.49)$$

La velocidad positiva en el aire causa una fuerza en elevación mayor sobre las palas, lo cual se manifiesta en un flap-back (vuelo hacia atrás) en el rotor;  $\frac{\partial(a_1)}{\partial(\mu_z)}$  se encarga de capturar este efecto. Un valor analítico se encuentra en [4, p. 159] para el vuelo hacia atrás, y es escalado por el coeficiente  $K_\mu$  para reflejar el efecto de la barra estabilizadora:

$$\frac{\partial(a_1)}{\partial(\mu_z)} = \frac{K_\mu 16 \mu^2}{(1 - \mu^2/2)(8 |\mu| + a\sigma)} \text{sign}(\mu) \approx K_\mu \frac{16 \mu^2}{8 |\mu| + a\sigma} \text{sign}(\mu). \quad (2.50)$$

#### 2.4.3.6. Fuerzas en el rotor

Para índices de avance pequeños ( $\mu < 0.15$ ), podemos asumir que el vector de empuje es casi perpendicular al disco del rotor (TPP). Para pequeños ángulos

de flapping (por debajo de los 10 grados) es posible usar aproximaciones lineales para determinar las componentes en los ejes principales del helicóptero. Para éste modelo el rotor esta conectado al fuselaje y por tanto las fuerzas en el rotor se trasladan directamente al centro de gravedad de todo el helicóptero, trayendo consigo su respectivo par que más adelante se verá:

$$X_{mr} = -T_{mr}a_1, \quad (2.51)$$

$$Y_{mr} = T_{mr}b_1, \quad (2.52)$$

$$Z_{mr} = -T_{mr}. \quad (2.53)$$

#### 2.4.4. Modelo del Motor, gobernador (*governor*) y velocidad del rotor

Antes de iniciar esta parte del capítulo, es importante comentar que el pitch de la pala esta estrechamente conectada a al torque suministrado por el motor del helicóptero. En el helicópteros que se utilizan usualmente para competencias entre aeromodelistas, el control el empuje se produce simultáneamente por un aumento o disminución del angulo de ataque de las hélices y de la cantidad de combustible inyectado. Esta combinación esta en modo lineal para ambos casos cuando el piloto es principiante. La combinación es cuadrática o cúbica de acuerdo a la destreza del piloto. La última combinación hace que el aumento de la fuerza de elevación sea brusca, logrando violentos despegues que son del disfrute de muchos de estos aeromodelistas. Para nuestro caso vamos a utilizar la combinación lineal es decir las variaciones simultáneas van a ser de carácter lineal ya que el indice de avance y la condición de hover son para movimientos relativamente lentos.

La dinámica del rotor puede ser modelada por la siguiente ecuación, según

[15, p. 153]:

$$\dot{\Omega} = \dot{r} + \frac{Q_e - Q_{mr} - n_{tr}Q_{tr}}{I_{rot}}. \quad (2.54)$$

Donde  $I_{rot}$  es el momento de inercia del las hélices y el eje de transmisión.  $Q_e$  es el torque del motor en el sentido de la agujas del reloj.  $Q_{mr}$  es torque en el rotor principal.  $n_{tr}$  es el indice de engrane del rotor trasero.  $\Omega$  es la velocidad del rotor principal.  $\dot{\Omega} - \dot{r}$  es la aceleración del rotor. El toque del motor depende de la cantidad del combustible inyectada al motor y la velocidad del rotor, la primera cantidad se representa por medio del control  $\delta_t$ . Esto se puede cuantificar por medio del uso de tablas o curvas características.

El CALIBER 5 está equipado con un motor de la marca japonesa O.S. MAX-50SX-HRING HYPER, cuyas características se muestran a continuación (ver tabla 2.2):

Tabla 2.2: Especificaciones del motor de combustion  
MAX-50SX-HRING HYPER

<b>Parámetro</b>	<b>Descripción</b>
Cilindrada (Displacement)	8.17 cc
Diámetro interno del pistón (Bore)	22.0 mm
Distancia recorrida por el pistón (Stroke)	21.5 mm
RPM	2000 – 20000 r.p.m
Potencia	1.87 hp
Peso	406 g

Sin embargo, es posible asumir que la potencia es proporcional a la entrada de Throttle ( $\delta_t$ ), la cual es proporcional a su vez a la cantidad de Nitrometano

(combustible para el Motor MAX-50SX-HRING HYPER) que ingresa a la cámara de combustión.

$$P_e = P_e^{\text{máx}} \delta_t, \quad (2.55)$$

donde  $0 < \delta_t < 1$ . Luego el torque del motor es

$$Q_e = \frac{P_e}{\Omega}. \quad (2.56)$$

De acuerdo con [5, p. 44] la respuesta en toque del motor puede ser considerada instantánea, debido a que los retrasos con el ingreso del aire, el flujo de combustible y combustión son muy pequeños comparados con la dinámica del vehículo.

Debido a que no existen datos respecto al motor MAX-50SX-HRING HYPER proporcionados por el fabricante. El governor va a ser implementado utilizando un controlador Proporcional-Integral (PI). De tal manera que se mantenga una velocidad constante del giro de la hélices del helicóptero. Esto es

$$\delta_t = K_p (\Omega_c - \Omega) + K_i \int (\Omega_c - \Omega) dt, \quad (2.57)$$

donde  $\Omega_c$  es la velocidad angular de referencia.  $K_p$  y  $K_i$  son las ganancias proporcional. Además la dinámica del servo motor es más rápida que la dinámica del del rotor, por lo que fue despreciada [5, p. 44]. Seguidamente el polinomio característico del sistema motor-governor es como sigue [5, p. 45]:

$$P(s) = s^2 + \frac{s(3Q_{mr}^o + P_e^{\text{max}} Kp)}{\Omega I_{rot}} + \frac{P_e^{\text{max}} K_i}{\Omega I_{rot}}. \quad (2.58)$$

Según (2.58) podemos deducir que:

$$2\zeta\omega_n = \frac{3Q_{mr}^o + P_e^{\text{max}} Kp}{\Omega I_{rot}}, \quad (2.59)$$

$$\omega_n^2 = \frac{P_e^{max} K_i}{\Omega I_{rot}}, \quad (2.60)$$

y

$$I_{rot} = 2 I_{\beta_{mr}} + I_{es} n_{es}^2 + 2 I_{\beta_{tr}} n_{tr}^2, \quad (2.61)$$

donde  $I_{\beta_{mr}}$  y  $I_{\beta_{tr}}$  son las inercias de las palas del rotor principal y de la cola,  $I_{es}$  es la inercia del eje del motor con todos los componentes rotatorios,  $n_{tr}$  y  $n_{es}$  se muestran en la tabla 2.1. La contribución más importante es el momento de inercia de las palas del rotor principal. El momento de inercia de la cola representa aproximadamente el 5% de la inercia del rotor principal. La inercia relacionada con el motor es difícil de calcular. Sin embargo, asumiendo una masa de 0.2 Kg y un radio efectivo de 0.04 m, tenemos que el momento de inercia  $I_{rot}$  es aproximadamente 2.5 veces la inercia de las palas del rotor principal [5, p. 46].

Como ya se ha visto, el control del gobernador nos va a proporcionar una velocidad constante en la velocidad de giro del rotor principal, ésta a su vez debe estar cercana a la velocidad nominal de 1770 RPM que se obtiene a partir de la tabla 2.1. No obstante, el incremento en el torque aerodinámico hace que la velocidad disminuya temporalmente y por tanto la velocidad de giro en las palas de cola; en consecuencia, el torque yawing sufriría una disminución que tendrá que ser compensada o corregida rápidamente. En los helicópteros R/C de aeromodelistas esto se logra mediante un dispositivo llamado Yaw Gyro que se encarga de mantener a la cola firme.

#### 2.4.5. Fuerzas en el fuselaje

El downwash del rotor (aire debajo del rotor) se ve desviado en movimiento forward o de costado. Esto crea una fuerza que se opone al movimiento. Estas

fuerzas llamadas de arrastre se pueden expresar de la siguiente forma [5, p. 47]:

$$V_\infty = \sqrt{u_a^2 + v_a^2 + (w_a + V_{imr})^2}, \quad (2.62)$$

$$X_{fus} = -\frac{1}{2} S_x^{fus} \rho u_a V_\infty, \quad (2.63)$$

$$Y_{fus} = -\frac{1}{2} S_y^{fus} \rho v_a V_\infty, \quad (2.64)$$

$$Z_{fus} = -\frac{1}{2} S_z^{fus} \rho (w_a + V_{imr}) V_\infty, \quad (2.65)$$

donde  $S_x^{fus}$ ,  $S_y^{fus}$  y  $S_z^{fus}$  son las áreas efectivas de arrastre frontal, de lado y vertical respectivamente,  $u_a$ ,  $v_a$  y  $w_a$  son la velocidades de los centros de presiones respecto del aire (para nuestro caso el centro de presiones coincide con el centro de gravedad).  $u_a = (u - u_w)$ , donde  $u_w$  es la velocidad del viento en la dirección  $X$  del cuerpo rígido. De acuerdo con [5, p. 47]  $S_y^{fus} \approx 2.2S_x^{fus}$  y  $S_z^{fus} \approx 1.5S_x^{fus}$ . El área frontal efectiva se puede determinar del ángulo pitch promedio requerido para mantener una cierta velocidad de avance. En el caso de hacer una prueba experimental, tenemos que  $mg\theta \approx -\frac{1}{2} S_x^{fus} \rho u_a^2$ . Un ángulo pitch de -10 grados fue presentado para una velocidad del viento de 14.5 m/s en la dirección  $X$ , lo cual resultó en  $S_x^{fus} = 0.1 \text{ m}^2$

#### 2.4.6. Rotor de la cola

El rotor de la cola esta ubicado en una posición donde existen una grand cantidad del condiciones de flujo, particularmente en el vuelo a baja velocidad [15, p. 142]. Además del inflow que se vio para el rotor principal también representa una dificultad. Vamos a trabajar el análisis del empuje en el rotor de la cola en condiciones de trim (condiciones de equilibrio vuelo con controles seteados como: hover, cruise o autorotation), donde el empuje del rotor nunca es cero para poder compensar el torque del rotor principal. Por tanto la ecuaciones (2.30) y (2.31)

son aplicables.

$$C_{T_{\mu_z^{tr}}}^{tr} = \frac{\partial (C_T^{tr})}{\partial (\mu_z^{tr})}, \quad (2.66)$$

$$C_{T_{\delta_r}}^{tr} = \frac{\partial (C_T^{tr})}{\partial (\delta_r)}. \quad (2.67)$$

Las ecuaciones (2.66) y (2.67) fueron computadas numéricamente. Los coeficientes resultantes adimensionales fueron usados para calcular las correspondientes derivadas dimensionales de estabilidad [5, p. 49]:

$$Y_v^{tr} = -\frac{C_{T_{\mu_z^{tr}}}^{tr} f_t \rho \Omega_{tr} R_{tr} \pi R_{tr}^2}{m}, \quad (2.68)$$

$$Y_{\delta_r}^{tr} = -\frac{C_{T_{\delta_r}}^{tr} f_t \rho (\Omega_{tr} R_{tr})^2 \pi R_{tr}^2}{m}, \quad (2.69)$$

donde  $f_t$  es el factor de obstrucción del fin, cuyo valor según [15, p. 143] es:

$$f_t = 1.0 - \frac{3}{4} \frac{S_{vf}}{\pi R_{tr}^2}. \quad (2.70)$$

La velocidad de la cola del rotor está dado por  $\Omega_{tr} = n_{tr} \Omega_{mr}$ . Finalmente, la fuerza lateral generada por el rotor de la cola está dada por:

$$Y_{tr} = m Y_{\delta_r}^{tr} \delta_r + m Y_v^{tr} \mu_z^{tr} \Omega_{tr} R_{tr}. \quad (2.71)$$

Con el objetivo de calcular  $Y_{tr}$ , necesitamos calcular las componentes de flujo normal  $\mu_z^{tr}$  y la de plano  $\mu^{tr}$ . Para modelar la forma compleja de como afecta el rotor principal al rotor de la cola. Vamos a introducir un factor wake del rotor principal  $K_\lambda$ . Calculamos los siguiente parámetros geométricos (tangentes de los ángulos que determinan la geometría):

$$g_i = \frac{l_{tr} - R_{mr} - R_{tr}}{h_{tr}}, \quad (2.72)$$

$$g_f = \frac{l_{tr} - R_{mr} + R_{tr}}{h_{tr}}. \quad (2.73)$$

Cuando el rotor de la cola esta totalmente inmerso  $K_\lambda = 1.5$ . En los otros casos cuando el rotor está parcialmente inmerso, se asume un crecimiento lineal del factor de intensidad de wake o estela (ver figura 2.8) para una velocidad de avance:

$$K_\lambda = 1.5 \frac{\left(\frac{u_a}{V_{imr} - w_a} - g_i\right)}{(g_f - g_i)}. \quad (2.74)$$

La expresión (2.74) es usada para determinar la componente vertical de la



Figura 2.8: Vortices wake en modo avance

velocidad del aire. Ahora calculamos el índice de avance para el rotor de la cola:

$$u_{tr} = \frac{u_a^2 + w_{tr}^2}{\Omega_{tr} R_{tr}}. \quad (2.75)$$

Donde la componente normal al rotor de la cola está dado por:

$$u_{tr} = v_a - l_{tr}r + h_{tr}p, \quad (2.76)$$

luego

$$\mu_z^{tr} = \frac{v_{tr}}{\Omega_{tr} R_{tr}}. \quad (2.77)$$

La magnitud el empuje en la cola está limitado por el máximo coeficiente de empuje, y está dado por la siguiente expresión:

$$Y_{max}^{tr} = f_t C_{T_{max}}^{tr} \rho (\Omega_{tr} R_{tr})^2 \pi R_{tr}^2 \geq |Y_{tr}|. \quad (2.78)$$

Finalmente el momento yawing y el pequeño momento rolling debido a la desviación del rotor de la cola respecto del centro de gravedad son:

$$N_{tr} = -Y_{tr} l_{tr}, \quad (2.79)$$

$$L_{tr} = Y_{tr} h_{tr}. \quad (2.80)$$

Sin embargo necesitamos calcular la velocidad inducida en el rotor de la cola, el cual será usado para calcular la fuerza vertical en el fin. Por tanto, así como se vio para el rotor principal. El índice de influjo es esta definido por la ecuación (2.81).

$$\lambda_0^{tr} = \mu_z^{tr} - 2 \left( \frac{2C_T^{tr}}{a_{tr} \sigma_{tr}} - \delta_r \left( \frac{1}{3} + \frac{\mu_{tr}^2}{2} \right) \right), \quad (2.81)$$

donde  $\sigma_{tr} = \frac{2c_{tr}}{\pi R_{tr}}$  es el índice de solidez del rotor de la cola,  $C_T^{tr}$  es el coeficiente de empuje del rotor de la cola y  $a_{tr}$  es la pendiente de la curva de empuje del rotor de la cola. Los parámetros restantes se encuentra en la tabla 2.1.

#### 2.4.7. Momentos y fuerzas en el fin vertical

Según [5, p. 47] la fuerza lateral producida por el fin vertical es

$$Y_{vf} = -\frac{1}{2} \rho S_{vf} \left( C_{L_\alpha}^{vf} V_\infty^{tr} + |v_{vf}| \right) v_{vf}, \quad (2.82)$$

donde  $S_{vf}$  es el área del fin vertical,  $C_{L\alpha}^{vf}$  es la pendiente de la curva de empuje,  $V_{\infty}^{tr} = \sqrt{u_a^2 + w_{tr}^2}$  es la velocidad axial ubicada en el plano del rotor de la cola.  $v_{vf}$  es la velocidad lateral relativa al aire ubicada en el fin vertical,  $w_{tr}$  es la velocidad vertical, ambos se muestran a continuación:

$$v_{vf} = u_a - \epsilon_{vf}^{tr} V_{itr} - l_{tr} r \quad (2.83)$$

y

$$w_{tr} = w_a + l_{tr} q - K_{\lambda} V_{imr}; \quad (2.84)$$

donde  $V_{itr}$  es la velocidad inducida en el rotor de la cola,  $r$  es la velocidad angular yaw,  $\epsilon_{vf}^{tr}$  es la fracción del fin vertical expuesta a la velocidad inducida del rotor de la cola,  $l_{tr}$  es la distancia vertical entre el centro de gravedad y el centro del rotor trasero, el cual se considera a la misma distancia del centro de presiones del fin vertical.

El valor absoluto de la fuerza lateral en el fin vertical, está dado por:

$$|Y_{vf}| \leq \frac{1}{2} \rho S_{vf} \left( (V_{\infty}^{tr})^2 + v_{vf}^2 \right). \quad (2.85)$$

Finalmente la fuerza lateral del fin vertical crea un momento yawing y un pequeño momento rolling debido a sus desviaciones respecto del centro de gravedad, y viene dado por:

$$N_{vf} = -Y_{vf} l_{tr} \quad (2.86)$$

y

$$L_{vf} = Y_{vf} h_{tr} \quad (2.87)$$

### 2.4.8. Fuerzas y momentos en el estabilizador horizontal

El estabilizador horizontal produce un fuerza de levantamiento y un además un momento pitching estabilizador alrededor del centro del gravedad [5, p. 48]. Asumiendo que la cola horizontal se encuentra totalmente sumergido en el downwash del rotor principal:

$$w_{tr} = w_a + l_{ht}q - K_\lambda V_{imr}. \quad (2.88)$$

Este mismo factor de intensidad es usado para fin horizontal como para el fin vertical y el rotor trasero. Luego la fuerza generada por el estabilizador horizontal está determinado por:

$$Z_{ht} = \frac{1}{2} \rho S_{ht} (C_{L_\alpha}^{ht} |u_a| w_{ht} + |w_{ht}| w_{ht}), \quad (2.89)$$

donde  $S_{ht}$  es área del estabilizador horizontal,  $C_{L_\alpha}^{ht} = 3.0$  es la pendiente de la curva de levantamiento. El valor absoluto de la fuerza de levantamiento de la barra horizontal está delimitado por:

$$|Z_{ht}| \leq \frac{1}{2} \rho S_{ht} (u_a^2 + w_{ht}^2). \quad (2.90)$$

Finalmente, el momento pitching generado por el estabilizador horizontal es

$$M_{ht} = Z_{ht} l_{ht}. \quad (2.91)$$

### 2.4.9. Ecuaciones integradas del helicóptero

Un modelo de simulación requiere de un orden de ocurrencia de los subsistemas de ecuaciones vistas en la presente sección. La figura 2.9 muestra un arreglo típico de como las fuerzas y momentos dependen del movimiento de la aeronave, los

controles y las perturbaciones atmosféricas.

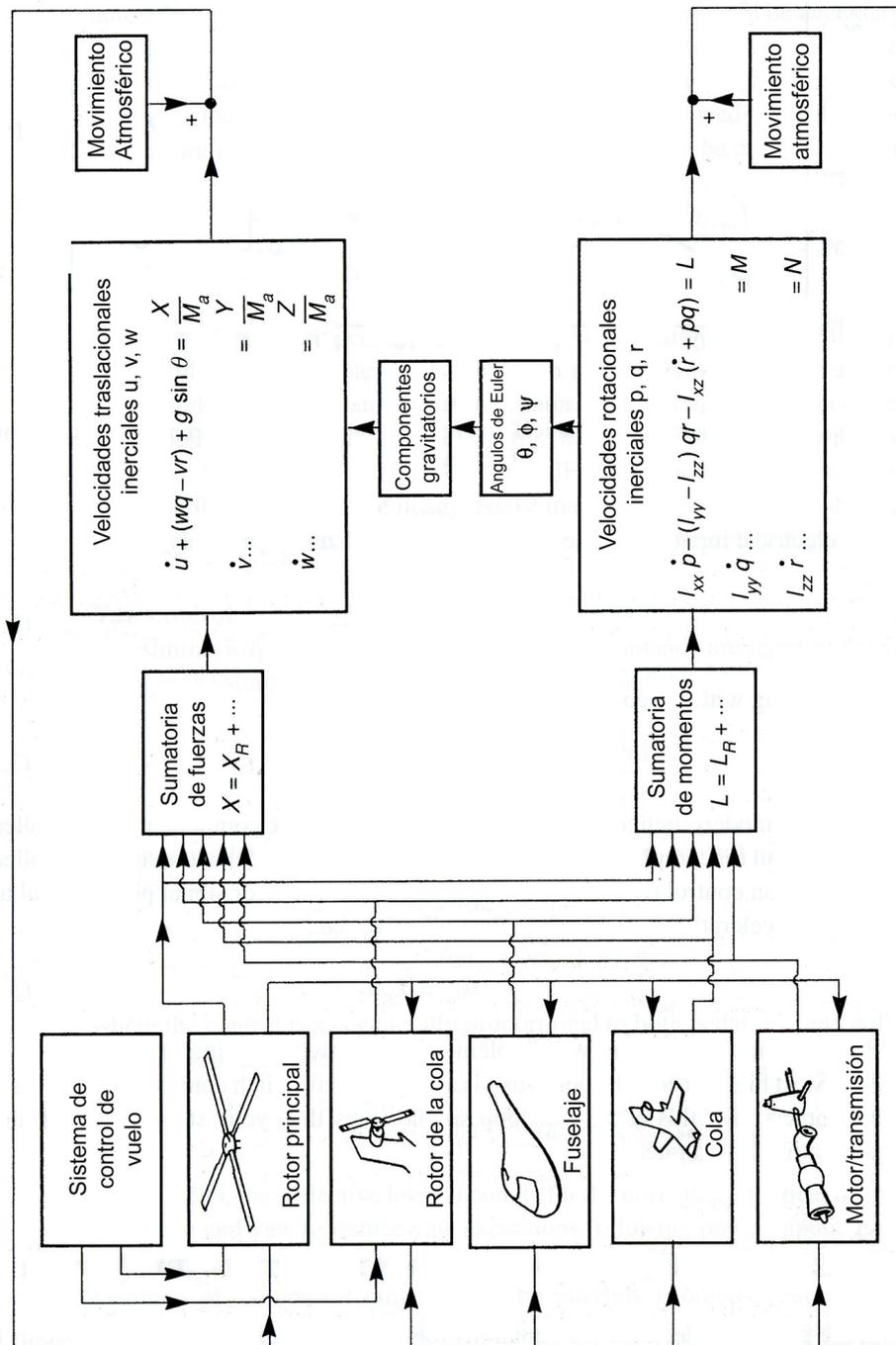


Figura 2.9: Modelo integrado de simulación del helicóptero.

## 2.5. Emparejamiento de barrido estándar

En ésta sección vamos a explicar los puntos más importantes del algoritmo de emparejamiento de barrido estándar o también conocido como *standard scan matching*. El algoritmo tiene como sensor principal al LADAR (Laser Detection and Ranging), el cual emite una cierta cantidad haces de luz en un determinado rango; por ejemplo, por una región de trabajo de  $270^\circ$ . Cada tiempo y velocidad de rebote de cada haz proporciona a la computadora del sensor los datos necesarios para calcular la distancia de separación de la fuente de emisión y el medio. De ésta forma, se puede tener información bidimensional de objetos distantes. La figura 2.10 muestra la forma cómo el haz de luz rojo al girar es capaz de registrar el contorno en dos dimensiones de la región encerrada en el marco verde. La información del contorno registrado es almacenado en un arreglo.

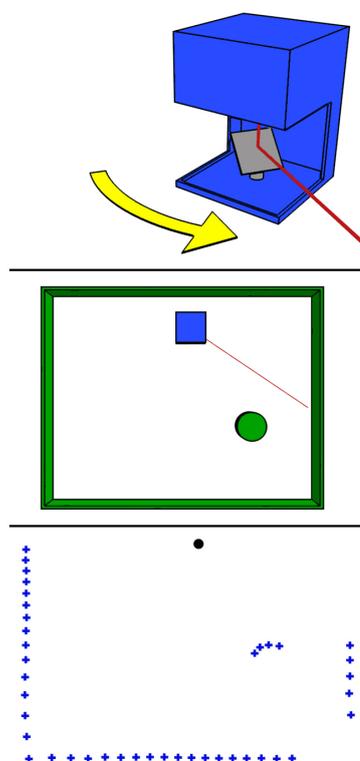


Figura 2.10: Funcionamiento general del sensor ladar.

El emparejamiento de barrido tiene la característica primordial de encontrar la relación de traslación más rotación que correlaciona dos nubes o grupos de puntos. Esta correlación encontrada permite encontrar los cambios de posición y orientación; en otras palabras, si de manera ilustrativa movemos el sensor láser y simultáneamente capturamos escaneos, es posible encontrar los cambios de posición y orientación entre dos barridos o escaneos consecutivos. El algoritmo que permite calcular una mejor relación de transformación entre dichos barridos es conocido como *iterative closest point* (punto más próximo iterativo).

### 2.5.1. Algoritmo *iterative closest point* (ICP)

El algoritmo ICP tiene como objetivo encontrar la transformación entre una nube de puntos y alguna superficie de referencia (o también otra nube de puntos), por medio de la minimización de los errores entre puntos correspondientes [3]. Además, éste algoritmo es viene del hecho de que las correspondencias son consideradas como una solución que más cercana al error local mínimo. Así como cualquier método de gradiente, el ICP es aplicable cuando tenemos de por sí un buen punto de comienzo [3]; en caso contrario, podría quedar atrapado en otro mínimo local que no representa la relación de transformación verdadera.

En el campo de la robótica, el ICP ha sido extensivamente empleado para emparejar escaneos de láser en 2D; también, denominado *scan matching*. Como podemos ver en la figura 2.11, el juego de puntos conocido como escena es alineado al juego de puntos llamado modelo. Esto se hace mediante una transformación de rotación más traslación. El proceso naturalmente es iterativo, por lo que podrían encontrarse varias iteraciones intermedias hasta encontrar las transformación acumulada óptima. Una forma de poder de poder encontrar el error de alineamiento entre dos nubes de puntos es mediante la suma de errores cuadráticos de los puntos correspondientes de los juegos de puntos. Sin embargo,

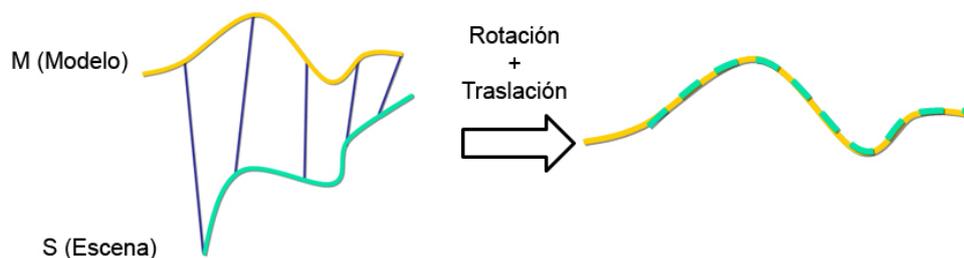


Figura 2.11: Visión general del estándar ICP.

¿cómo encontramos los puntos correspondientes?. Una forma muy efectiva y rápida es mediante la distancia más corta entre dos puntos; es decir, encontramos para cada punto en el juego de puntos modelo su puntos más cercano en el juego de puntos escena, dichos puntos serán correspondientes. Esto se ilustra claramente en la figura 2.12. Si las correspondencias correctas son encontradas, podemos

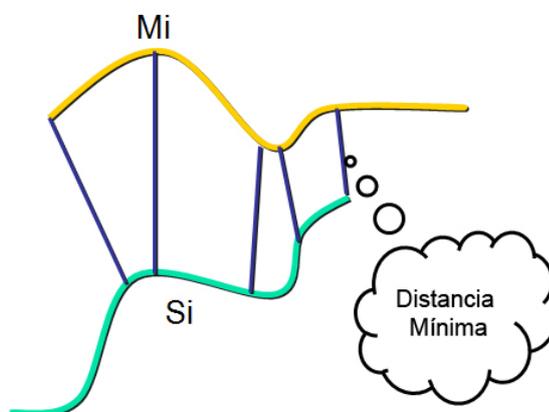


Figura 2.12: Consideración de correspondencia del punto más cercano.

minimizar el error de alineamiento y hallar finalmente la rotación y traslación óptima. Seguidamente, en la figura 2.13 podemos apreciar el diagrama de flujo del algoritmo de emparejamiento de barrido. En ella se muestran los pasos del algoritmo ICP. En el paso de “cálculo del alineamiento” se procede a estimar la rotación y traslación de 3 puntos tanto en el juego de puntos modelo y en el de escena. En la ecuación 2.92 se puede ver el procedimiento para estimar la

relación de transformación entre los tres puntos y sus correspondientes.  $X_m$  y  $Y_s$  son las matrices compuestas por tres puntos en el modelo y tres puntos de la escena respectivamente. Cada punto, a su vez, está compuesto de coordenadas  $x$  y  $y$ . Y finalmente  $T_{rot+tras}$ , que luego de ser despejada, representa la relación de transformación entre las mencionadas nubes de puntos.

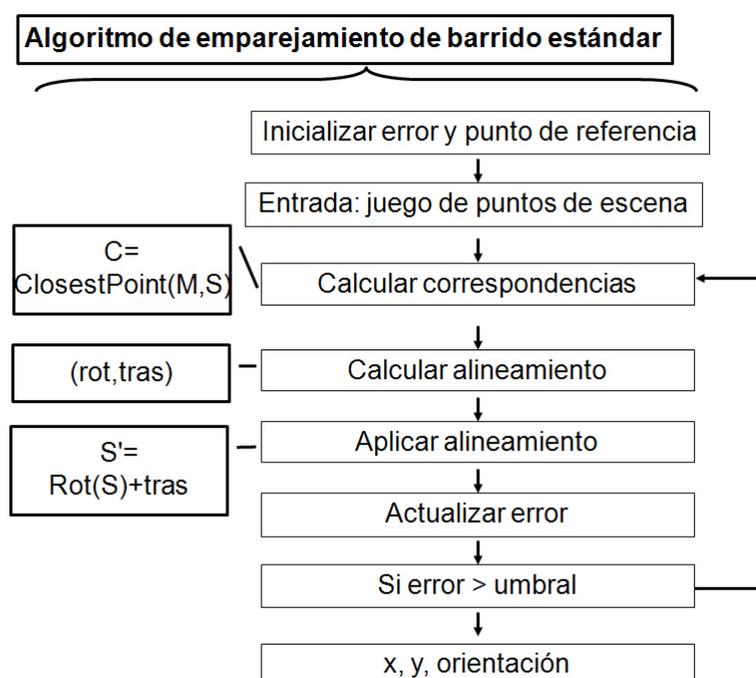


Figura 2.13: Diagrama de flujo del algoritmo de emparejamiento de barrido.

$$\begin{bmatrix} x_m & x_{2m} & x_{3m} \\ y_m & y_{2m} & y_{3m} \\ 1 & 1 & 1 \end{bmatrix} = [T_{rot+tras}] \begin{bmatrix} x_s & x_{2s} & x_{3s} \\ y_s & y_{2s} & y_{3s} \\ 1 & 1 & 1 \end{bmatrix} \quad (2.92)$$

Nuevamente en la figura 2.13, la expresión  $ClosestPoint(M, S)$  representa la función de distancia más corta entre los puntos del modelo y la escena. El paso de “aplicar alineamiento”, se procede a aplicar las relaciones de transformación a la escena; es decir, después de éste paso la nube de puntos escena estará más alineada a la nube de puntos modelo. El proceso se repite hasta que el error de

alineación esté por debajo del umbral requerido.

De ésta forma, el algoritmo estándar de emparejamiento de barrido queda sustentado por medio del algoritmo ICP (Iterative Closest Point). Posteriores modificaciones y mejoras serán explicadas en los subsiguientes capítulos.

## 2.6. Conceptos básico de probabilidad

En la robótica probabilística, las variables de los sensores, de control y estados se asumen como aleatorias [20, p. 10]. Denotemos a  $X$  como una variable aleatoria y  $x$  como un evento específico que  $X$  podría tener. Si el espacio de datos de  $X$  es discreto, la probabilidad de que la variable aleatoria  $X$  tenga un valor de  $x$  es:

$$p(X = x) \tag{2.93}$$

Por ejemplo, una moneda tirada al aire tiene dos posibilidades de ocurrencia  $p(X = \textit{cara}) = 1/2$  o  $p(X = \textit{sello}) = 1/2$ . Y éstas probabilidades discretas suman uno.

$$\sum_x p(X = x) = 1 \tag{2.94}$$

Además una probabilidad siempre es positiva ( $0 \leq p(X = x)$ ). Para simplificar la notación mostrada en 2.93, es común usar la abreviación  $p(x)$  envés de  $p(X = x)$ .

Muchas técnicas de estimación y toma de decisiones trabajan en espacios continuos, es decir, sus variables pueden tomar valores continuos [20, p. 10]. Además todas las variables aleatorias continuas poseen funciones de densidad de probabilidad. Por ejemplo, la más usada es la distribución normal con media  $\mu$  y varianza o covarianza  $\sigma^2$ . Esta distribución está dada por la siguiente expresión:

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right) \tag{2.95}$$

## 2.7. Predictor de Kalman

En primer lugar vamos a presentar la definición de filtro de Kalman. Este método matemático fue nombrado así en honor a Rudolf E. Kalman (ver figura 1.6). El propósito del filtro es usar mediciones con ruido (variaciones aleatorias) u otras imprecisiones, que son observadas en el tiempo, y producir así una tendencia de valores que se aproximen a los verdaderos valores de los datos medidos. El filtro de Kalman es ampliamente usado en aeronáutica e ingeniería por dos razones importantes: por combinar las mismas variables, pero de diversos sensores y por combinar predicciones inexactas de los estados del sistema con las medidas inexactas de dichos estados [16, p. 1]. Además algunas extensiones y generalizaciones se han desarrollado como en el caso de [11].

El filtro de Kalman representa la esperanza por los momentos de representación [20, p. 34]. En el tiempo  $t$ , la esperanza es representada por la media  $\mu_t$  y la covarianza  $\Sigma_t$ . Las siguientes serán Gaussianas, si se mantienen las siguientes tres propiedades, además de las condiciones de Markov del filtro de Bayes [20, p. 34].

1. La siguiente probabilidad de estado  $p(x_t|u_t, x_{t-1})$  debe ser una función lineal en sus argumentos con el ruido Gaussiano añadido. Esto se expresa en el siguiente ecuación:

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \quad (2.96)$$

Donde  $x_t$  y  $x_{t-1}$  son vectores de estado.  $u_t$  es el vector de control en el tiempo  $t$ . Ambos vectores podemos considerarlos verticales y los mostramos en la expresión 2.97.

$$x_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{bmatrix} \wedge u_t = \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ u_{3,t} \end{bmatrix} \quad (2.97)$$

$A_t$  y  $B_t$  son matrices.  $A_t$  es una matriz de  $n \times n$ , donde  $n$  es la dimensión del vector de estado  $x_t$ .  $B_t$  es de tamaño  $n \times m$ , donde  $m$  es la dimensión del vector de control  $u_t$ . Al multiplicar el vector de estados y el vector de control con las matrices  $A_t$  y  $B_t$  respectivamente, la función de estados transitorio se convierte en lineal en sus argumentos [20, p. 35].

La variable aleatoria  $\epsilon_t$  en 2.96 es un vector aleatorio Gaussiano que modela la aleatoriedad en el estado transitorio. Su media es cero y su covarianza será representada por  $R_t$ .

Aplicando la definición de distribución normal mostrada en la ecuación 2.98, tenemos que la media del estado siguiente es  $A_t x_{t-1} + B_t u_t$  y la covarianza es  $R_t$ .

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (2.98)$$

$$p(x_t | u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right) \quad (2.99)$$

2. La probabilidad de medición  $p(z_t | x_t)$  tiene también que ser lineal en sus argumentos, con el ruido Gaussiano añadido:

$$z_t = C_t x_t + \delta_t \quad (2.100)$$

Donde  $C_t$  es una matriz de tamaño  $k \times n$ , donde  $k$  es la dimensión del vector de medición  $z_t$ . El vector  $\delta_t$  describe la medición del ruido. La distribución de  $\delta_t$  es un Gaussiano multivariable con media cero y covarianza  $Q_t$ . La media de la probabilidad está, en consecuencia, dada por la siguiente distribución normal multivariable:

$$p(z_t | x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)\right) \quad (2.101)$$

3. Finalmente, la esperanza inicial  $bel(x_0)$  debe ser normalmente distribuida.

Denotaremos la media de ésta esperanza por  $\mu_0$  y su covarianza como  $\Sigma_0$ .

$$bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1} (x_0 - \mu_0)\right) \quad (2.102)$$

Estas tres condiciones son suficientes para asegurar que la esperanza posterior a  $bel(x_t)$  sea siempre una Gaussiana para cualquier instante en el tiempo.

### 2.7.1. El algoritmo del filtro de Kalman

El algoritmo de Kalman está mostrado en la tabla 2.3. El filtro de Kalman representa la esperanza  $bel(x_t)$  en el tiempo  $t$ , con media  $\mu_t$  y covarianza  $\Sigma_t$ . La entrada del filtro de Kalman es la esperanza en el tiempo  $t - 1$ , representada por  $\mu_{t-1}$  y  $\Sigma_{t-1}$ . Para actualizar éstos parámetros, el filtro de Kalman requiere del control  $u_t$  y de la medición  $z_t$ . La salida es la esperanza en el tiempo  $t$ , representada por  $\mu_t$  y  $\Sigma_t$ . En las líneas 1 y 2 de la tabla 2.3, la esperanza predicha

Tabla 2.3: Algoritmo del filtro de Kalman para estados de transición y mediciones Gaussianas lineales

<b>Algoritmo Filtro Kalman</b> ( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):	
1	$\bar{\mu}_t = A_t x_{t-1} + B_t u_t$
2	$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
3	$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
4	$\mu = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
5	$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
6	return( $\mu_t, \Sigma_t$ )

para  $\mu$  y  $\Sigma$  es calculada representando la esperanza  $\bar{bel}_t$  en el instante de tiempo posterior, pero antes de la incorporación de la medición  $z_t$ . Esta esperanza es obtenida por la incorporación de  $u_t$ . La media es actualizada utilizando la versión determinística de la función de transición en la ecuación 2.96, con la media  $\mu_{t-1}$  sustituida por el estado  $x_{t-1}$ . La actualización de la covarianza considera el hecho

que los estados dependen de los estados anteriores a través de la matriz lineal  $A_t$ , que es multiplicada dos veces con la covarianza, ya que ésta es una matriz cuadrática [20, p. 36].

La esperanza  $\overline{bel}(x_t)$  es subsecuentemente transformada en la esperanza deseada  $bel(x_t)$  en las líneas 3, 4 y 5, por medio de la incorporación de la medición  $z_t$ . La variable  $K_t$ , computada en la línea 3 y es llamada ganancia de Kalman. La línea 5 manipula la media, por medio del ajuste de la misma en proporción a la ganancia de Kalman  $K_t$  y la desviación de la medición actual  $z_t$ , y la medición predicha de acuerdo a la relación 2.100. Finalmente, la nueva covarianza de la esperanza posterior es calculada en la línea 5, ajustando también la ganancia de información que resulta de la medición [20, p. 37].

## 2.7.2. El predictor de Kalman

El predictor de Kalman trabaja de forma análoga a lo visto en el desarrollo del algoritmo del filtro de Kalman; sin embargo, hay una simple consideración que lo hace predictor. Si en el caso anterior los datos de ingreso y los de salida estaban en los instantes de tiempo  $t - 1$  y  $t$  respectivamente, en éste caso, los instantes de tiempo serán  $t$  y  $t + 1$ . Para ello consideramos el predictor desarrollado por Karlsson et al. [11, p. 5]. El algoritmo se muestra en la tabla 2.4.

Tabla 2.4: Algoritmo del predictor de Kalman

<b>AlgoritmoPredictorKalman</b> ( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):	
1	$e_t = z_t - C_t \bar{\mu}_t$
2	$K_t = A_t \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
3	$\mu_{t+1} = A_t \bar{\mu}_t + K_t e_t$
4	$\bar{\Sigma}_{t+1} = (A_t - K_t C_t) \bar{\Sigma}_t (A_t - K_t C_t)^T + R_t + K_t Q_t K_t^T$
5	return( $\mu_{t+1}, \Sigma_{t+1}$ )

# Capítulo 3

## Diseño del Software de Simulación de Vuelo

### 3.1. Diseño de entorno en realidad virtual

En el mundo actual los simuladores han tomado un papel preponderante en el diseño de sistemas mecánicos. Mediante una serie de procesos computacionales es posible visualizar el comportamiento de los sistemas en cuestión. Existen muchas plataformas de simulación y animación. Todas ellas tienen particularidades en común; por ejemplo, muchas de esas plataformas tienen librerías en común como es el caso de las librerías `glui.h` y `glaux.h`. Las librerías anteriores son las denominadas librerías de OpenGL.

OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que construyan gráficos 2D y 3D. La interfaz consiste de unas 250 funciones diferentes, las cuales permiten dibujar escenas tridimensionales complejas a partir de figuras geométricas primitivas, tales como puntos, líneas, triángulos, etc. Se desarrolló por Silicon Graphics Inc. en 1992 y se usa en CAD, realidad virtual, representación científica entre otros. Asimismo, es posible encontrarla en el desarrollo de video

juegos. Una de las aplicaciones concernientes con el diseño del simulador es la posibilidad de simular ecuaciones variables en el tiempo. Debido a que esta plataforma corre en lenguaje C, es posible insertar las ecuaciones de movimiento para que en todo momento sean resueltas. La plataforma automáticamente reproduce los comandos insertados en el código y muestra una visualización del sistema mecánico en movimiento en dos y tres dimensiones.

Para dar una visión general del desarrollo del simulador de vuelo del helicóptero vamos a dividirlo en tres partes: Ajuste de entorno 3D, Inserción del helicóptero 3D e Inserción de la ecuaciones mecánicas.

### 3.1.1. Ajuste de entorno 3D

El ajuste 3D consiste en ajustar las condiciones externas de animación, tales como iluminación, posición del foco de luz, color de la luz de iluminación, suavizado, entre otros.

El extracto de código que se muestra a continuación tiene por objetivo mostrar las características más importantes de de la configuración del entorno. Como en todo entorno siempre se limpia la pantalla, esto se hace con el comando `glClearColor(0.0, 0.0, 0.0, 0.0)`. El comando `glEnable()` activo el Depth Buffer, el cual es usado para la eliminación de superficies ocultas. `glEnable()` activa el cálculo de iluminación. `glEnable()` activa la fuente de luz. `glEnable()` activa el coloreo de material. `glShadeModel()` agrega una sombra suave. Para mayor detalle visitar <http://www.opengl.org/>.

```

1 glClearColor (0.0 , 0.0 , 0.0 , 0.0);
2 glEnable (GL_DEPTH_TEST);
3 glEnable (GL_LIGHTING);
4 glEnable (GL_LIGHT0);
5 glEnable (GL_COLOR_MATERIAL);

```

6 `glShadeModel(GLSMOOTH);`

Una vez que ya se han detallado de manera general las características del entorno, lo siguiente es conocer el procedimiento de funcionamiento del programa. Para esto es importante detallar el el programa principal, pues aquí se van ejecutando todas la funciones que componen al programa.

En primer lugar, el programa principal se ejecuta constantemente. Aunque no tiene una función *while()* que repita el programa principal. Este cuenta con una estructura interna que hace que la función principal continúe repitiendose de manera indefinida. Aclarado este punto vamos a detallar cada comando que muestra debajo. La línea *glutInit()* inicializa la librería glut.h y además este comando recibe dos parámetros que son interpretados por el main. El siguiente comando *glutInitDisplayMode()* configura el modo de video inicial, y recibe en este caso tres parámetros con los que indica la configuración deseada. El comando *glutInitWindowSize(800,650)* indica el tamaño de la ventana de ejecución en pixeles. *glutInitWindowPosition(100,50)* indica la coordenada en pixeles del extremo superior izquierdo de la ventana de ejecución. El comando *glutCreateWindow()* le asigna una nombre a la ventana de ejecución. La siguiente línea llama al a función *initRendering()*. El comando *glutDisplayFunc(drawScene)* llama a la función *drawScene()*, la cual se encarga de dibujar en la ventana de ejecución el entorno tridimensional. El comando *glutKeyboardFunc(handleKeypress)* se encarga de llamar la función *handleKeypress()*, la cual captura información de que tecla se ha presionado para realizar una instrucción dependiendo de cual tecla fue. *glutSpecialFunc(specialFunc)* actúa de manera similar, sólo que en éste caso se recibe información de la teclas especiales como las flechas directrices que aparecen en los teclados. El comando *glutReshapeFunc(handleResize)* se encarga de actualizar o refrescar la ventana de ejecución, por ejemplo cuando la ventana

es arrastrada, éste comando de encarga de refrescar la animación a la posición final después del arrastre. El comando *glutIdleFunc(control)* contiene todas la ecuaciones dinámicas del simulador. El comando *glutMainLoop()* se encarga de hacer que la función principal se repita indefinidamente. Finalmente, *return 0* retorna cero al final de la función principal. Para mayor referencia visitar <http://www.opengl.org/>.

En resumen, la librería OpenGL es apropiada para el simulador del helicóptero, debido a que no solamente proporciona un entorno tridimensional de animación, sino que también proporciona una plataforma de fácil acceso o entrada de ecuaciones dinámicas.

```

1  glutInit(&argc , argv );
2  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
3  glutInitWindowSize(800, 650);
4  glutInitWindowPosition(100, 50);
5  glutCreateWindow("SOFTWARE_DE_VUELO" );
6  initRendering ();
7  glutDisplayFunc(drawScene);
8  glutKeyboardFunc(handleKeyPress);
9  glutSpecialFunc(specialFunc);
10 glutReshapeFunc(handleResize);
11 glutIdleFunc(control);
12 glutMainLoop();
13 return 0;

```

Hasta ahora se ha detallado la estructura más importante del Simulador HeliSi3D (**Helicopter Simulator 3D**), es decir los comandos principales que hacen que el simulador anime el entorno tridimensional del helicóptero. Algunos de estos comandos pueden ser prescindibles por ejemplo el comando

*glutInitWindowPosition(100, 50)*, pues si se obviara ésta sentencia el programa trabajaría con la posición por defecto. Sin embargo, los otros comandos como *glutDisplayFunc(drawScene)* le dan la vida al simulador ya que sin este comando no se podría visualizar simplemente nada. A continuación vamos a detallar a grandes rasgos las funciones mas importantes del HeliSi3D.

### 3.1.2. Función *initRendering()*

La función *initRendering()* está encargada básicamente de dos tareas: la primera de configuración y la segunda de cargado de imagen base del piso. Estas tareas van a configurar el entorno tridimensional del HeliSi3D con aspecto de textura, color e iluminación, así como también la imagen de lo que nos daría a entender de un piso. A continuación se muestra el código de la función *initRendering()*.

```

1 //CONFIGURAMOS LAS CARACTERISTICAS DE RENDERIZADO
2 glClearColor(0.0, 0.0, 0.0, 0.0);
3 glEnable(GL_DEPTH_TEST);
4 glEnable(GL_LIGHTING);
5 glEnable(GL_LIGHT0);
6 glEnable(GL_NORMALIZE);
7 glEnable(GL_COLOR_MATERIAL);
8 glShadeModel(GL_SMOOTH);
9 //CARGAMOS LA IMAGEN DEL PISO
10 Image* image = loadBMP("square.bmp");
11 _textureId = loadTexture(image);
12 delete image;
```

La primera labor como se mencionó líneas arriba es la de configurar y preparar el entorno de animación. Para lograr este objetivo es importante limpiar la

pantalla y eso lo hacemos mediante el comando *glClearColor(0.0, 0.0, 0.0, 0.0)* (que está en el extracto de programa mostrado arriba), el mencionado comando ya se explicó en la sección de ajuste de entorno 3D. Seguidamente, los comando de iluminación y textura le van a dar el aspecto tridimensional y ciertamente natural al simulador. Por ejemplo las luces principales son activadas mediante los comandos *glEnable(GL\_LIGHTING)* y *glEnable(GL\_LIGHT0)*. Además se activa la forma de incidencia de la luz en forma normal mediante el comando *glEnable(GL\_NORMALIZE)*. Dicho comando permite que la figuras dentro del simulador puedan ser visualizadas de manera natural. Finalmente el comando *glEnable(GL\_COLOR\_MATERIAL)* le da la tonalidad al entorno 3D, el cual es suavizado mediante el comando *glShadeModel(GL\_SMOOTH)*. Finalmente, todos los comandos reunidos prepararán el ambiente de animación para la inserción de sólidos o cuerpos tridimensionales.

La segunda labor de la función *initRendering()* es de insertar el piso del entorno de animación. Todos los sólidos insertados dentro del entorno 3D no tienen restricciones de colisión, es decir, los cuerpos dentro de ambiente de animación no van a responder a choques ni colisiones. En el caso del piso, éste tampoco va a responder a restricciones físicas. Por lo tanto lo que se ve es simplemente una imagen de un espesor mínimo, el cual da la idea de un piso. Por ejemplo, si un cuerpo 3D se mueve a través del piso o plano, éste podrá atravesarlo ya que no hay ecuaciones de restricciones físicas en el programa de animación. Regresando al código que se muestra arriba, la variable puntero *image* es creada y cargada con la imagen "square.bmp" la cual se puede apreciar en la figura (3.1). Seguidamente se carga la textura de imagen en la variable *\_textureId*, mediante el comando *loadTexture(image)*. Finalmente se borra la variable *imagen* con el comando *delete*.

El resultado visible se muestra en la figura (3.2). Para lograr este resultado hacen falta todavía pasos siguientes; sin embargo, el resultado básicamente será el

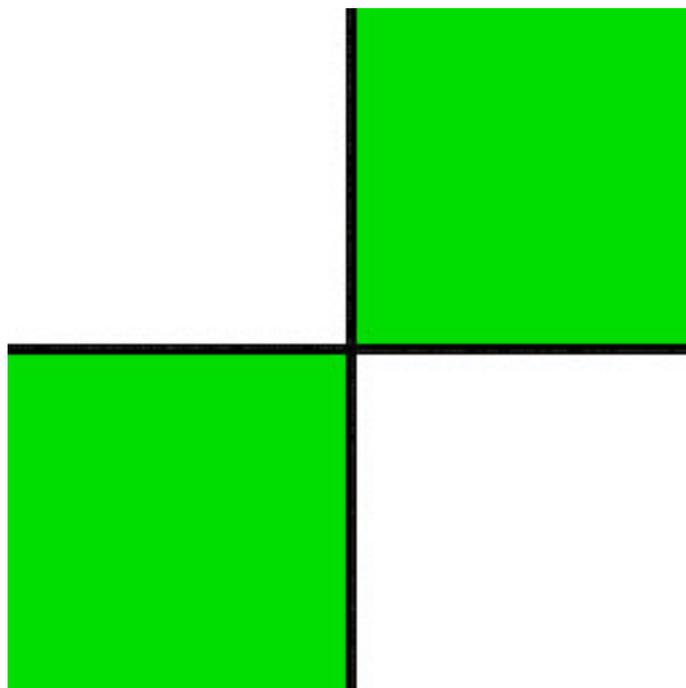


Figura 3.1: Imagen base del piso del HeliSi3D (tamaño=256x256 Pixeles)

mismo. El efecto de perspectiva que se puede apreciar claramente está incluido dentro de la librería `openGL`, es decir, éste paso ya viene preconfigurado. El paso que no se ha especificado y que también salta a la vista es de la repetición de la imagen piso base. Lo que se hace para que parezca un piso "infinito" es la multiplicación del piso base alrededor de una cierta posición inicial. Dicha multiplicación se hace de manera sucesiva en la direcciones  $xy$ . El ancho final del piso es decir la cantidad de imágenes piso base es de 1000 unidades, lo cual como resultado da la idea de un piso indefinido.

### 3.1.3. Función `drawScene()`

La función `drawScene()` es argumento del comando `glutDisplayFunc()` como se muestra en el extracto de código de la función principal. A continuación mostramos parte del código de la `drawScene()`.

```
1 //CONFIGURACIÓN INICIAL DEL ENTORNO
```

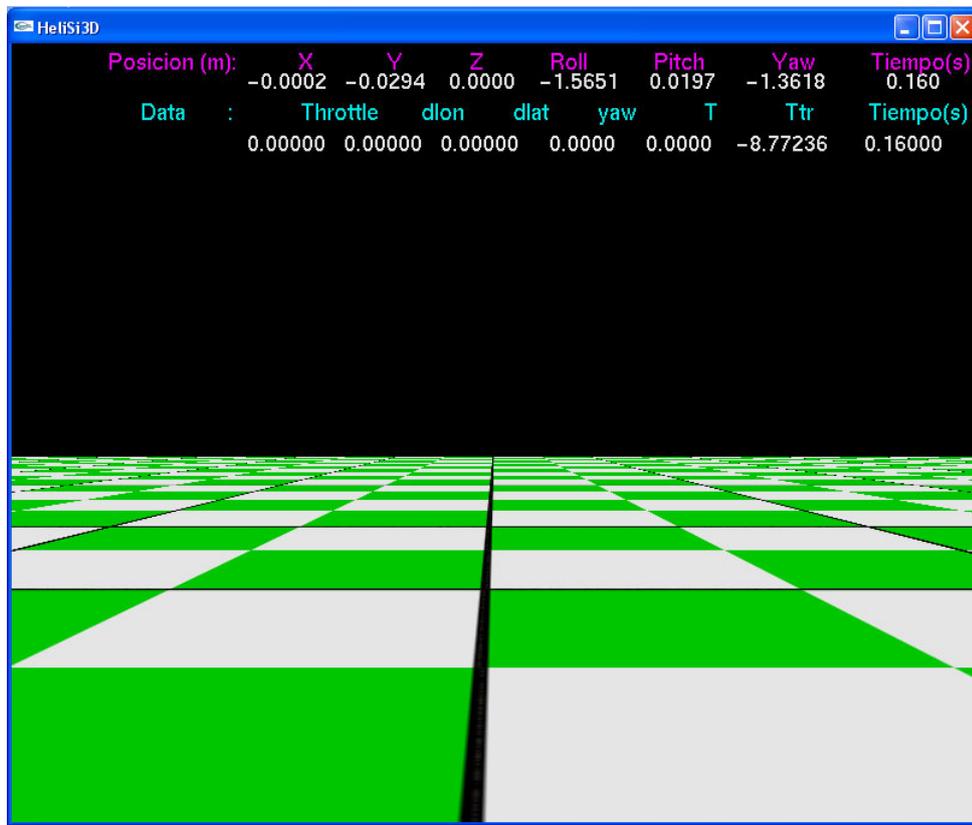


Figura 3.2: Piso del entorno de animación

```

2      glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
3      glMatrixMode (GL_MODELVIEW);
4      glLoadIdentity ();
5      //POSICION INICIAL DE CAMARA CON SEGUIMIENTO
6      glTranslatef (0.0 f+10*y, -5.0 f+10*z, -40.0 f+10*x);
7      glTranslatef (-derecha, 0.0 f, 0.0 f);
8      glTranslatef (0.0 f,-arriba, 0.0 f);
9      glTranslatef (0.0 f,0.0 f,-fondo);
10     //SETEAMOS INTENSIDAD DE LUZ DEL AMBIENTE
11     GLfloat ambientLight [] = {0.7 f, 0.7 f, 0.7 f, 1.0 f};
12     glLightModelfv (GL_LIGHT_MODEL_AMBIENT, ambientLight);
13     //SETEAMOS COLOR Y POSICION DE LA LUZ

```

```

14     GLfloat lightColor [] = {0.7f, 0.7f, 0.7f, 1.0f};
15     GLfloat lightPos [] = {-0.2f, 0.3f, -1, 0.0f};
16     glLightfv(GL_LIGHT0, GL_DIFFUSE, lightColor);
17     glLightfv(GL_LIGHT0, GL_POSITION, lightPos);
18     //DIBUJAMOS EL HELICOPTERO
19     glPushMatrix();
20         glColor3f(0.0, 0.0, 1.0);
21         glTranslatef(-10*y, -10*z, -10*x);
22         glRotatef(-180/PI*psi, 0.0f, 1.0f, 0.0f);
23         glRotatef(180/PI*theta, 1.0f, 0.0f, 0.0f);
24         glRotatef(-180/PI*phi, 0.0f, 0.0f, 1.0f);
25         glScalef(10.0f, 10.0f, 10.0f);
26     if (!scene.LoadFile(fus))
27     {
28         fprintf(stderr, "No_hay_archivo:_%s", fus);
29         exit(1);
30     }
31     Dibujar3DS();
32     glPopMatrix();
33     //CARGAMOS LA TEXTURA
34     glEnable(GL_TEXTURE_2D);
35     glBindTexture(GL_TEXTURE_2D, _textureId);
36     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
37     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
38     //CONFIGURAMOS LA DIMENSION DEL PISO
39     glBegin(GL_QUADS);
40     glNormal3f(0.0f, 1.0f, 0.0f);

```

```

41 glTexCoord2f(0,0);
42 glVertex3f(-2000.0f, 0.0f, -2000.0f);
43 glTexCoord2f(200,0);
44 glVertex3f(-2000.0f, 0.0f, 2000.0f);
45 glTexCoord2f(200,200);
46 glVertex3f(2000.0f, 0.0f, 2000.0f);
47 glTexCoord2f(0,200);
48 glVertex3f(2000.0f, 0.0f, -2000.0f);
49 glEnd();
50 glDisable(GL_TEXTURE_2D);
51 glutSwapBuffers();

```

En la parte inicial de “CONFIGURACIÓN INICIAL DE ENTORNO” seteamos las condiciones de inicio y borramos la pantalla. Seguidamente seteamos la posición inicial de la cámara con seguimiento. Hay dos entradas para modificar la posición de la cámara. La primera entrada proviene de la posición de simulación. Esto se observa en la sentencia de código *glTranslatef(0.0f + 10 \* y, -5.0f + 10 \* z, -40.0f + 10 \* x)*, es decir cada vez que la posición de simulación cambie, la cámara sigue automáticamente a la nueva posición del cuerpo o sólido. En segundo lugar tenemos la entradas provenientes del teclado. Las teclas (configuradas en la función *specialFunc(int key, int x, int y)*, para mayor información visitar <http://www.opengl.org/>) utilizas son los cuatro botones direccionales y los botones “z” y “x”. Naturalmente, los botones direcciones desplazan la cámara en la direcciones *xy* con respecto a la pantalla del monitor y las teclas “z” y “x” mueven la cámara en la dirección *z* dándole así la apariencia de alejamiento o de acercamiento. En la parte “SETEAMOS LA INTENSIDAD DE LUZ” configuramos la tonalidad de la luz que afectará al sólido dentro del entorno, es decir este comando no afecto al entorno configura anteriormente. De igual

manera en “SETEAMOS COLOR Y POSICIÓN DE LA LUZ” configuramos el color de la luz incidente en el sólido, así como también a posición focal de la luz incidente. Por ejemplo, se sabe que la librería OpenGL contempla una fuente de luz focal ubicada en el infinito y un punto por el cual éste pasa, dicho punto se especifica en  $GLfloatlightPos[] = -0.2f, 0.3f, -1, 0.0f$ . Luego tenemos “DIBUJAMOS EL HELICOPTERO” donde se especifica el color de la textura del sólido, la posición de simulación, las rotaciones de simulación y la escala a la cual se ajusta el helicóptero 3D. Para poder entender a profundidad el funcionamiento correcto de cada comando visitar <http://www.opengl.org/>. Por otro lado tenemos “CARGAMOS TEXTURA” donde cargamos la variable textura *\_textureId* dentro del entorno 3D; además, se le aplica un filtro de textura lineal como se muestra en el código. Finalmente, se ingresan los parámetros de piso. La forma de ingresar imágenes cuadrangulares es mediante el comando *glBegin(GL\_QUADS)*, seguidamente se especifica el vector unitario de la dirección de incidencia de la luz sobre el piso. Luego la forma de ingresar la imagen de piso base especificando su escala y cantidad es como se muestra en “CONFIGURAMOS LA DIMENSION DEL PISO”. El comando *glutSwapBuffers()* es para alimentar los buffers del motor de animación con los datos de la nueva trama y así dar una aspecto continuo al entorno 3D durante la animación.

### 3.1.4. Función *handleKeypress()*

La función *handleKeypress()* es su vez argumento de la función *glutKeyboardFunc()*. Esta función contiene los comandos que se encargan de interaccionar el teclado y el software de animación. A continuación se muestra parte del código:

```

1 switch (key) {
2 case 27: //Boton escape
```

```

3         exit (0);
4         break ;
5     case 's' :
6     case 'S' :
7         psir +=10.0*PI/180.0;
8         break ;}

```

Como se puede apreciar en el extracto de código, aquí podemos configurar que la tecla escape aborta la animación, mediante el comando *exit(0)*. Además, con los botones 's' o 'S' podemos incrementar el valor de la variable *psir* cada vez que se actualize una trama, que se puede entender también como cada ocurrencia de periodo de muestreo.

### 3.1.5. Función *specialFunc()*

La función *specialFunc()* es el argumento del comando *glutSpecialFunc()*. Esta función desempeña un papel parecido a *handleKeyPress()*, con la diferencia que aquí solo se tratan los botones especiales como las teclas direccionales. La función desarrollada se muestra a continuación, donde podemos apreciar que es idéntica en estructura de *handleKeyPress()*.

```

1  switch (key) {
2  case GLUT_KEY_LEFT:
3      derecha -=1.0f ;
4      break ;
5  case GLUT_KEY_RIGHT:
6      derecha +=1.0f ;
7      break ;
8  case GLUT_KEY_UP:
9      fondo -=1.0f ;

```

```

10         break ;
11     case GLUT_KEY_DOWN:
12         fondo += 1.0f ;
13         break ;}

```

### 3.1.6. Función *handleResize()*

La función *handleResize()* la cual es asu vez argumento de la función *glutReshapeFunc()* está encarga de capturar la nueva posición de la subventana de animación. Esto se puede visualizar cuando se mueva la ventana principal del entorno hacia una nueva posición respecto de la pantalla del monitor. Por tanto esto permitirá que imagen continua se adapte a esa nuevas condiciones y proporcione una animación continua y sin distorsiones. El código se muestra a continuación. Cabe señalar también que esta parte del código es genérica, es decir, las líneas de código mencionadas están disponibles libremente en la web y son prácticamente de la misma forma para cualquier aplicación donde se requieran (Para mayor información visitar <http://www.opengl.org/>).

```

1     glViewport(0, 0, w, h);
2     glMatrixMode(GL_PROJECTION);
3     glLoadIdentity();
4     gluPerspective(45.0, (float)w / (float)h, 1.0, 200.0);

```

### 3.1.7. Función *control()*

La función *control()* es también argumento del comando *glutIdleFunc()*. Esta función es una de las más importantes, pues es aquí donde encontramos todas las ecuaciones dinámicas y aerodinámicas, y condiciones iniciales. Por tanto ésta función es la más extensa. Lo que contiene básicamente son la ecuaciones descritas

en el capítulo 2. Contiene una primera parte que se refiere la inicializaciones de todas la variables de control como ángulos de Euler y rotaciones de Euler. Y en la segunda parte contiene la ecuaciones dinámicas en forma discreta. Esta discretización fue realizada mediante el método de Euler a un periodo de muestreo de 5ms (cuatro veces más rápido que la velocidad de respuesta de los actuadores). Se seleccionó este periodo de muestreo debido principalmente a que la velocidad de respuesta máxima de los servomotores se da cada 20 milisegundos, y teniendo en cuenta también que la velocidad de respuesta del helicóptero es mucho más lenta a la repuesta de los actuadores; por tanto, el periodo de muestreo es justificado. A continuación mostramos parte del código de la función *control()*.

```

1 //EQUACIONES DINAMICAS: NEWTON EULER
2 u=(double)u+dt*(double)(v*r-w*q-g*sin(theta)+(X)/m);
3 v=(double)v+dt*(double)(w*p-u*r+g*sin(phi)*cos(theta)+(Y)/m);
4 w=(double)w+dt*(double)(u*q-v*p+g*cos(phi)*cos(theta)+(Z)/m);
5 p=(double)p+dt*(double)(q*r*(Iyy-Izz)/Ixx+(taux)/Ixx);
6 q=(double)q+dt*(double)(p*r*(Izz-Ixx)/Iyy+(tauy)/Iyy);
7 r=(double)r+dt*(double)(p*q*(Ixx-Iyy)/Izz+(tauz)/Izz);
8 { ... }
9 //EMPUJE
10 Vimr=sqrt(m*g/(2.0*PI*rho*Rmr*Rmr));
11 mu=sqrt((u-uw)*(u-uw)+(v-vw)*(v-vw))/(omegamr*Rmr);
12 muz=(w-ww)/(omegamr*Rmr);
13 sigma=2.0*cmr/(PI*Rmr);
14 CTmax=0.0055;
15 { ... }
16 T=CTideal*rho*(omegamr*Rmr)*(omegamr *Rmr)* PI*Rmr*Rmr;
17 //TORQUE

```

18  $Q=CQ*rho*(omegamr*Rmr)*(omegamr *Rmr)* PI*Rmr*Rmr*Rmr;$   
 19  $\{ \dots \}$

Para poder entender de manera más clara la función *control()*, vamos a desarrollar una breve descripción de la solución numérica de ecuaciones diferenciales, la cuáles en su mayoría representan a la ecuaciones dinámicas y aerodinámicas del modelo no lineal del Helicóptero.

### 3.1.8. Solución Numérica de Ecuaciones Diferenciales

Está claro del capítulo 2 que la ecuaciones del Modelo del Helicóptero son no lineales, la cuales están sujetas a determinadas señales de entrada. Un solución analítica está fuera de toda concepción racional, y en cambio los métodos numéricos sugieren una rápida y versátil salida para la solución de estas ecuaciones. En general los sistemas físicos se comportan de manera lineal; por lo tanto, las derivadas de sus variables de estados van a existir, y las series de expansión de Taylor pueden ser usadas para predecir éste comportamiento que no es sino la solución de dichas ecuaciones [18, p. 172].

La solución numérica de una trayectoria continuo implica que dada una condición inicial  $X(t_0)$  y una entrada de control  $U(t)$ , se puede calcular la secuencia discreta de valores de

$$X(t_0 + kT), \quad k = 1, 2, \dots \quad (3.1)$$

lo cual satisface a las ecuaciones de estado

$$\dot{X}(t) = f(X(t), U(t)). \quad (3.2)$$

Esto también es conocido como el *problemadelvalorinicial*, donde T es el periodo de muestreo. Una de la formas para encontrar las soluciones numéricas es

mediante el uso del método de Euler.

### 3.1.8.1. Método de Euler

Considérese el problema de valor inicial de la ODE más simple; un sistema de ecuaciones diferenciales de primer orden cuyas condiciones de contorno son la siguientes.

$$\frac{dx}{dt} = f(x, t), \quad x(t_0) = x_0. \quad (3.3)$$

El problema de encontrar una solución discreta de valores tiene una obvia relación con la serie de Taylor de orden 1 o Euler:

$$x(t_0 + T) = x(t_0) + T\dot{x}(t_0). \quad (3.4)$$

Esta formula no es muy precisa, a menos que se utilicen periodos de muestreo muy pequeños, el cual es parte de nuestro caso. Esto podría ser mejorado con el uso del método de Runge-Kutta, pero eso demandaría de un aumento significativo de esfuerzo computacional.

Finalmente, estamos listos para visualizar el resultado (ver figura 3.3) de poner todo en conjunto y armar el simulador. Debido a que el objetivo de la presente tesis no es ilustrar el desarrollo de simuladores de realidad virtual, se ha explicado este paso de manera somera; sin embargo, se debe enfatizar que cualquier profundización del tema se puede encontrar en la siguiente dirección electrónica <http://www.opengl.org/>. La figura 3.3 muestra también parámetros de simulación, como posiciones, ángulos de Euler, tiempo de simulación y señales de entrada. Dichas señales se actualizan en todo momento de acuerdo a la simulación. Para ilustrar la forma como se programó el mostrado de la variables internas del simulador, presentamos a continuación un extracto del código:

```
1 char *nombre=" Posicion (m):X" ;
```

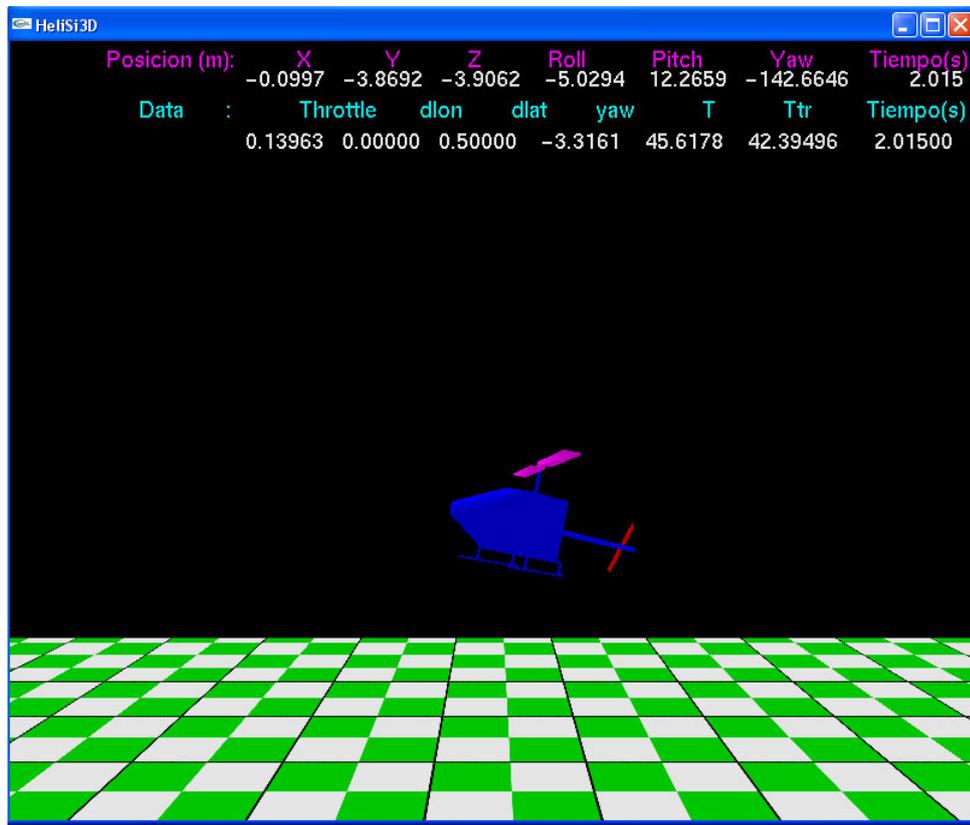


Figura 3.3: Simulador HeliSi3D

```

2  glColor3f(1.0,0.0,1.0);
3  sprintf(label," %s", nombre);
4  glRasterPos3f(-6.1f, 5.8f,-15.0f);
5  drawString(label);
6  glColor3f(1.0,1.0,1.0);
7  sprintf(label," %24.4f,x);
8  glRasterPos3f(-6.1f,5.5f,-15.0f);
9  drawString(label);

```

El código mostrado arriba permite mostrar el valor en todo instante de la variable de posición  $x$ . En primer lugar creamos una variable puntero nombre para el texto que deseamos mostrar que en esta caso es “Posicion (m):X”, en segundo lugar asignamos nombre a *label*; seguidamente, movemos lo asignado a la posición

deseada mediante el comando `glRasterPos3f()`; finalmente, mostramos el texto con el comando `drawString(label)`. Análogamente lo hacemos para mostrar el valor de posición  $x$ ; sin embargo ahora incluimos una ligera variación en la posición de traslación, es decir, tenemos que mostrar  $x$  en una posición por debajo del texto anterior displayado. Además, se incluye el cambio de color para poder diferenciar los valores numéricos del texto. El acabado final se puede apreciar en la figura 3.3.

Notar que los números dentro del programa incluyen una  $f$  de *float*, esto se debe a que el OpenGL esta estructurado para que reciba valores flotantes dentro de sus comandos de animación. Es decir, si tenemos variables *double* existe la necesidad de transformarlos a variables flotantes para poder ser mostradas en la pantalla apropiadamente.

### 3.1.9. Resultados del Simulador

En esta sección, vamos a presentar un resumen de los resultados del simulador HeliSi3D para nuestro minihelicóptero. Para poder mostrar comportamientos gráficos o resultados gráficos, se tuvo que hacer un programa similar al hecho en Visual C++. Dicho programa fue escrito en un M-file, el cual contó con características similares al código en C base. Sin embargo, el tratamiento de algunas variables tuvo un importante impacto en los resultados numéricos. Al inicio se trabajo con variables definidas en *float* en Visual C++, pero a medida que el tiempo de simulación avanzaba se iba perdiendo precisión con respecto a los valores obtenidos en Matlab. Esto se debió a que el Matlab trabaja la variables definidas en *double* y el Visual perdía precisión por su carácter flotante. Más tarde se uniformizó la definición en ambos programas y por ende se obtuvieron los mismos resultados numéricos. Una forma práctica de poder contrastar valor fue mediante la comparación de los parámetros mostrados en la pantalla del

simulador y los mostrados en la figura 3.4.

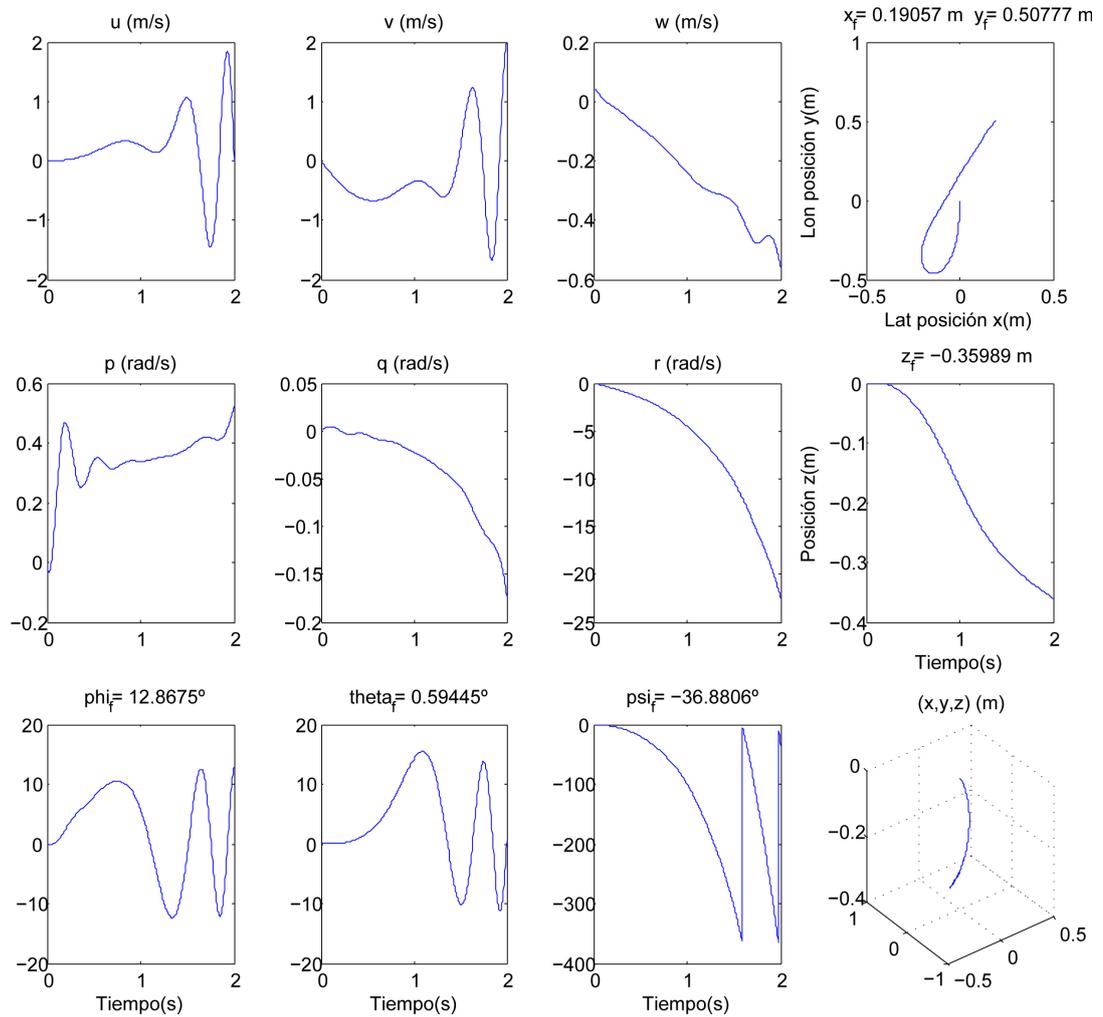


Figura 3.4: Comportamiento gráfico del simulador HeliSi3D para un ángulo de ataque en el rotor principal de  $\Theta = 4.0^\circ$ , un ángulo de ataque en el rotor de la cola de  $\Theta_{tr} = -10.0^\circ$ , una inclinación del swashplate de 50% en la dirección lateral, y una posición inicial ubicada en el origen de coordenadas. El subíndice  $f$  indica que el valor corresponde a la cantidad de la coordenada final de la gráfica.

## Capítulo 4

# Diseño del Sensor de Posición y Orientación Basado en Emparejamiento de Barrido Láser

Como de vio en el marco teórico, el algoritmo ICP (*iterative closest point*) es capaz de determinar el cambio de posición de nubes de puntos correspondientes, una respecto de la otra, en éste caso. Este cambio de posición puede ser acumulado de tal manera que se tenga la posición de un punto referencial (posición del sensor láser) en todo instante de tiempo. Además, éste algoritmo ha sido usado ampliamente en robots terrestres, los cuales se mueven en el plano. En éstos casos, el algoritmo ha sido aplicado de manera eficiente y efectiva ya que los robots terrestres no tienen límite de peso, pudiendo llevar computadoras de alta velocidad de procesamiento o un sinnúmero de implementos consigo. No obstante, los vehículos aéreos tienen una deficiencia en el peso límite que pueden levantar. Es por eso, que están restringidos a usar computadoras ligeras de bajos recursos, así como también, sensores livianos de baja potencia de alcance. Pese a ello, los

vehículos aéreos cuentan con dos importantes ventajas: su maniobrabilidad y su movilidad.

En la figura 4.1 podemos apreciar el algoritmo ICP y las modificaciones que se hicieron para hacer del mismo más robusto y efectivo. Las modificaciones que se muestran en la parte izquierda de la figura son: rechazo de barrido, búsqueda en la vecindad, predicción de movimiento y MPP (most probable pose technique) o técnica de la pose más probable, donde la pose viene dada por la posición y la orientación en el plano. Estas modificaciones serán explicadas detalladamente a continuación.

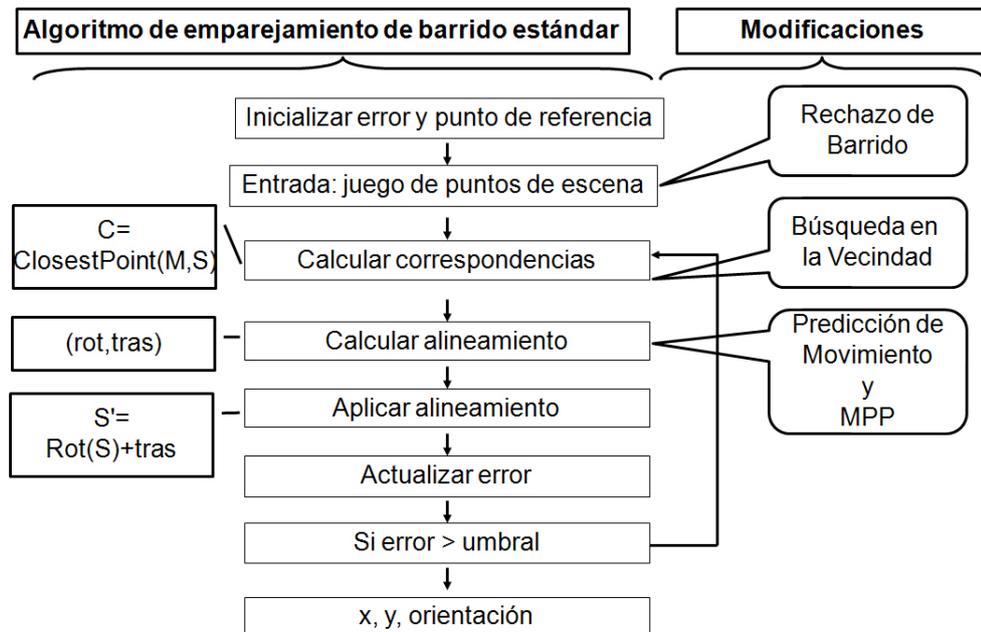


Figura 4.1: Algoritmo ICP y modificaciones. MPP (most probable pose technique) o técnica de la pose más probable.

## 4.1. Búsqueda en la vecindad

En el marco teórico se señaló que una forma de encontrar correspondencias entre dos juegos de puntos es mediante la relación del punto más cercano. Una

forma de encontrar el punto más cercano del juego de puntos referencia al juego de puntos escena es por medio de la minimización de las distancias de éste punto en el referencia a todos los puntos de la escena, tal como se muestra en la figura 4.2. Sin embargo, éste método es computacionalmente costoso, pues requiere de muchas comparaciones para calcular sólo una correspondencia. En otras palabras es de complejidad  $O(N^2)$ , cuadrática de tamaño  $N$ . Por eso es necesario utilizar un método que reduzca el costo computacional, para lo cual utilizamos el método de “búsqueda en la vecindad”.

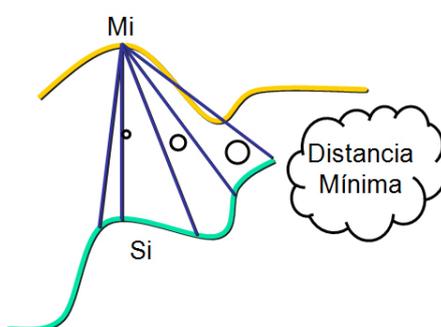


Figura 4.2: Determinación de la distancia más corta de un punto en la referencia  $M_i$  a la escena  $S_i$ .

Jost et al. ?? revisa varios métodos de los existentes y a la vez propone otros para reducir el costo computacional del algoritmo ICP. Este es un método que se basa en un enfoque heurístico para encontrar la mínima distancia entre los puntos de dos arreglos de puntos. En la figura 4.3, podemos ver el boceto general de la técnica de búsqueda en la vecindad, la cual es resumida de forma siguiente:

- Entradas: juego de puntos  $M$  y  $S$ , los cuales están asociados a sus vecindades:  $V(M)$  y  $V'(S)$  respectivamente.
- Salidas: para cada  $M_i$  de  $M$ , existe una aproximación  $S_i$  de su punto más cercano en el juego de puntos  $S$ .



de Kalman visto es la sección de filtro de Kalman de fundamento teórico.

El predictor de Kalman se usó para dar un tanteo inicial a la matriz de transformación y además para proveer de una error de convergencia inicial para el algoritmo ICP. El cambio de pose estimado por el predictor se usa para reacomodar la nube de puntos de la escena, es decir se la aplica una matriz de transformación basada en el cambio de posición estimado por el predictor. Luego, se calcula el error de alineamiento, el cual será usado como el umbral máximo de error. Los subsiguientes errores que vengan de en la iteraciones serán considerados siempre y cuando sean menores a éste umbral máximo. Ambas consideraciones pueden ser visualizadas en la figura 4.4.

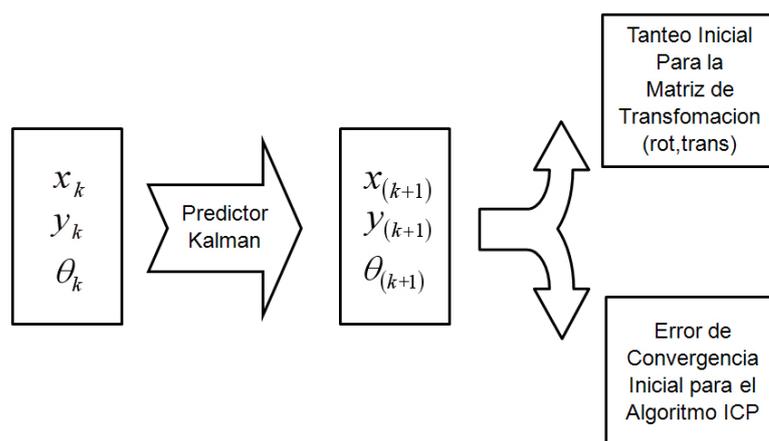


Figura 4.4: Filtro de Kalman es configurado para estimar  $x$ ,  $y$  y orientación. Esto permite tener dos cosas: un tanteo inicial para la matriz de transformación y un error convergencia inicial para el algoritmo ICP.

### 4.3. Técnica de la pose más probable

Luego de la primera estimación de movimiento por el predictor de Kalman, las posteriores matrices de transformación son encontradas de manera aleatoria. Los posibles cambios de pose son estimados según la relación 2.92. Calculamos un cierto número de estimaciones (en nuestro fue de aproximadamente cien), las

cuales son almacenadas y sometidas a un proceso de discriminación, es decir, sólo los cambios de pose más probables serán considerados. Además, se calcula la media y desviación estándar de éstas estimaciones. Cabe señalar que todas éstas predicciones representan el mismo cambio de pose, sin embargo, muchas de ellas, debido al carácter aleatorio de su cálculo, podrían estar fuera de un contexto físico posible. Por ejemplo, el cambio de posición entre dos instantes de tiempo (100 ms aproximadamente) no puede ser mayor a 10 cm, puesto que la velocidad máxima de desplazamiento del helicóptero quadrotor es de 1 m/s. Por lo tanto, todas las predicciones que estén fuera de éste límite, ciertamente aerodinámico, serán automáticamente desechadas.

Las relaciones que resumen el proceso de descarte de predicciones vienen expresadas en la siguiente forma:

$$\mu_{\Delta\theta} - \sigma_{\Delta\theta} < \Delta\theta(M_i, S_i) < \mu_{\Delta\theta} + \sigma_{\Delta\theta}, \quad (4.1)$$

$$\mu_{\Delta x} - \sigma_{\Delta x} < \Delta x(M_i, S_i) < \mu_{\Delta x} + \sigma_{\Delta x}, \quad (4.2)$$

$$\mu_{\Delta y} - \sigma_{\Delta y} < \Delta y(M_i, S_i) < \mu_{\Delta y} + \sigma_{\Delta y}. \quad (4.3)$$

En la figura 4.5 podemos apreciar cómo las estimaciones de la orientación pueden ser aproximadamente acomodadas en la campana de Gauss. Esto de manera explicativa, a fin poder visualizar la exclusión de estimaciones que se encuentren relativamente alejadas de las estimaciones más probables. Además de manera análoga, la figura 4.5 sirve para ilustrar los casos de  $x$  y  $y$ . Seguidamente se procede a formar un arreglo con todas las posibilidades que cumplan las condiciones 4.1, 4.2 y 4.3. De éstas posibilidades solo el caso que minimice el error de alineamiento será considerado como el cambio de pose más probable. Seguidamente se dará paso a subsiguientes iteraciones en la cuales el error de alineamiento de partida será el error mínimo de la iteración anterior de tal

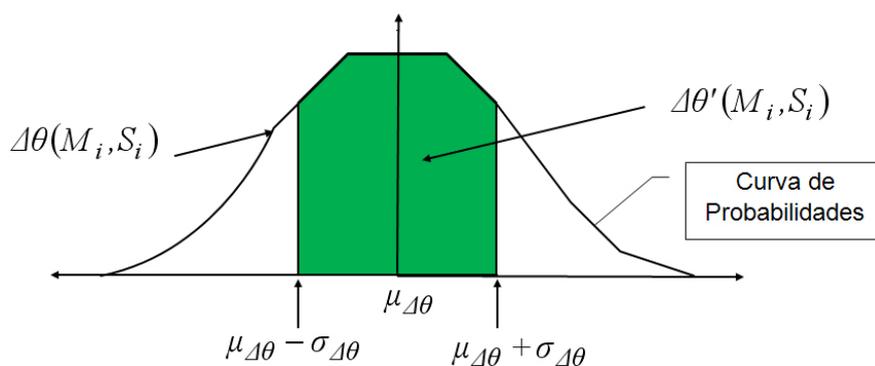


Figura 4.5: Las estimaciones de orientación  $\theta$  pueden ser hipotéticamente acomodadas en la campana de Gauss; a fin de mostrar la exclusión de estimaciones alejadas de las estimaciones más probables.

manera que se garantice que el error vaya convergiendo hasta mínimo posible; sin embargo, si el error durante una determinada iteración es menor que el umbral o error mínimo de alineamiento, entonces el proceso iterativo acaba. Finalmente, el cambio de pose más probable es almacenada y así da inicio a la recepción del siguiente juego de puntos escena.

#### 4.4. Rechazo de barrido basado en la actitud

Durante excursiones reales con el helicóptero, se detectó la ineludible presencia de grandes ángulos de pitch o roll. Esto podría deberse a la presencia de una corriente de viento, frenado repentino, o cambio de dirección de movimiento. Cuando ésto ocurre, el sensor láser que venía escaneando en un plano horizontal súbitamente cambia de plano de trabajo a uno subyacente o suprayacente al plano original. Consecuencia directa de ésto es que la información proveniente de un plano que no corresponde al horizontal podría traer consigo información (e.g. del suelo) que no podrá ser alineada con respecto al barrido modelo o de referencia. Además esto podría causar propagación excesiva del error y mal estimación de pose. Sin embargo, para ángulos de pitch o roll cercanos a cero, el algoritmo

podrá realizar la alineación con un error aceptable. Por tanto, los ángulos alejados de éste umbral (e.g.  $5^\circ$ ) causarán que los barridos que vengan de estas situaciones de actitud (roll y pitch) sean automáticamente rechazados como barridos modelo, mas no como barridos escena. De tal manera que no acumulen error, pero que sí proporcionen un valor aproximado del cambio de pose actual. El algoritmo se muestra continuación:

```

1  if(pitch < p_threshold || roll > r_threshold)
2    Barrido escena es rechazado.
3  endif

```

En la parte superior de la figura 4.6, podemos ver al helicóptero quadrotor el cual está sobrevolando de manera manual. En la parte inferior de la misma figura podemos ver que el barrido de escena es descartado por que su ángulo, porque su ángulo roll asociado al plano de trabajo instantáneo es menor a  $-5^\circ$ , siendo dicho valor el mínimo permisible para realizar el emparejamiento de barrido. Cabe señalar que el quadrotor aparentemente está volando horizontalmente, por eso es que muchas veces éstos ángulos de pitch y roll son despreciados, pero que al fin y al cabo influyen en la calidad de los resultados.

## 4.5. Optimizaciones

Se hicieron muchas consideraciones en el software para poder lograr el resultado óptimo.

En la técnica de búsqueda en la vecindad, se consideró un pequeña área de búsqueda en la vecindad del barrido modelo ( $V(M_i)$ ), pero se consideró una región de búsqueda relativamente grande en el barrido de referencia, a fin de poder considerar el caso en que el punto más cercano a un cierto punto en el modelo se encuentre con mayor certeza en la vecindad de barrido de escena ( $V(S_i)$ ).

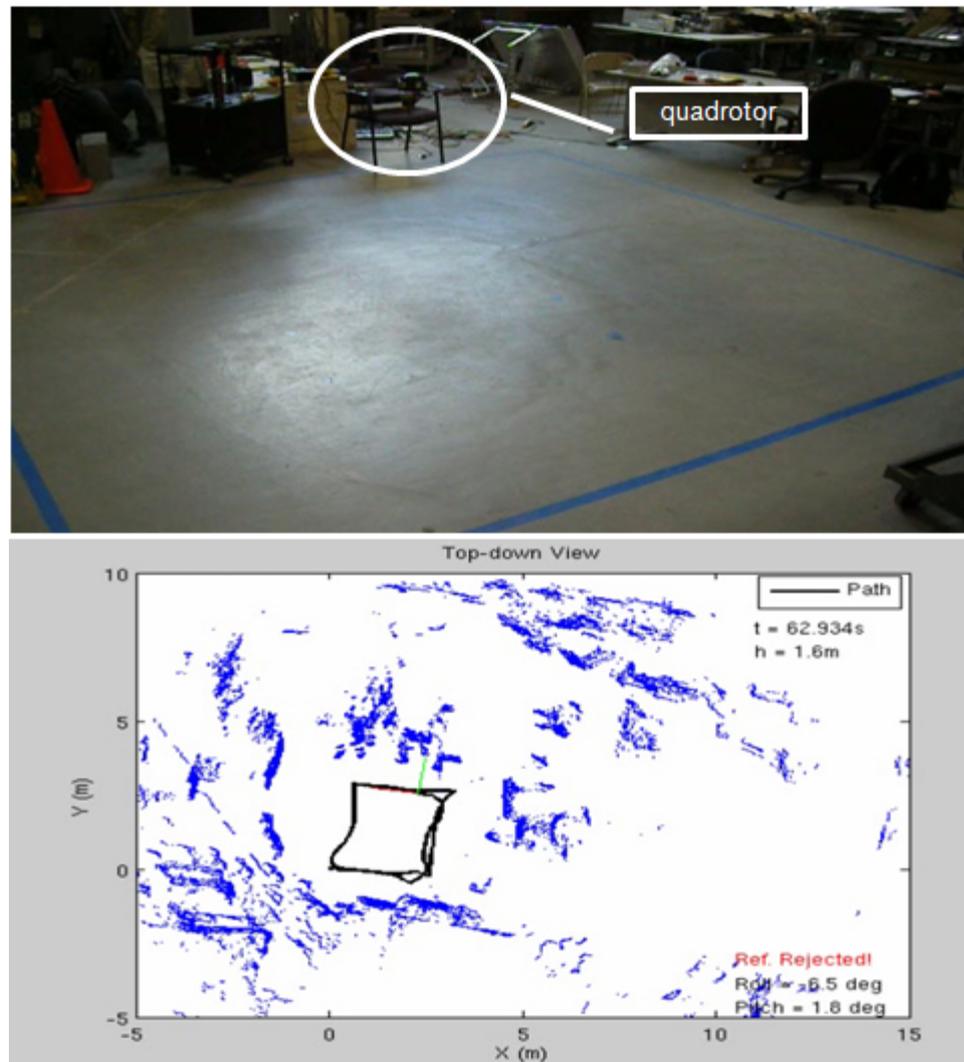


Figura 4.6: En la figura de arriba, el quadrotor aparece equipado y sobrevolando en modo manual. En la figura de abajo, vemos que el barrido de referencia que corresponde a ese instante de tiempo es rechazado debido a que su ángulo roll es menor a  $-5^\circ$ .

En la parte de predicción de movimiento se consideraron parámetros de varianza de acuerdo a velocidad de respuesta de la planta, así como también al resultado obtenido. Se obtuvo al final resultados relativamente pequeños, con lo cual se logró una respuesta rápida y precisa.

En la parte de la técnica de la pose más probable, se trabajó en zonas de hasta  $\pm 1.5\sigma$  por encima y debajo de la media  $\mu$ , con lo cual se logró agilizar el cálculo del cambio de posición más probable.

Finalmente, se determinó que los rangos de rechazo por encima de  $4^\circ$  o  $5^\circ$ , protegían al algoritmo contra información pobre para el emparejamiento de barrido. Como se comentó anteriormente, ésta consideración es necesaria para salvaguardar las estimaciones de posición y orientación ya que en misiones reales, los helicópteros de escala pequeña están expuestos a muchas perturbaciones que causan un movimiento caótico de largos ángulos de roll y pitch.

# Capítulo 5

## Control de Posición de un Mini Vehículo Aéreo

### 5.1. Arquitectura de Software

En ésta sección vamos a detallar el proceso de implementación del algoritmo de estimación de posición y el diseño del controlador. Ambos algoritmos fueron implementados en la computadora del quadrotor perteneciente al laboratorio de microvehículos aéreos del instituto de robótica de la universidad Carnegie-Mellon.

#### 5.1.1. Implementación del algoritmo de estimación de pose

El algoritmo de estimación de pose fue implementado íntegramente en C++. La plataforma utilizada fue Ubuntu 7.04 y fue implementada en un pico-computador VIA EPIA PX10000G Pico-ITX (para mayor información, visitar <http://www.logicsupply.com/products/px10000g>). Lo importante de usar una pico-computadora es su liviano peso y sus periféricos de comunicación, a diferencia de los DSP o los FPGA que necesitan de implementar protocolos

de comunicación y peor aun el soporte es exclusivo y por ende costoso, éstas computadoras ya vienen con todos los periféricos de una computadora convencional (mouse, teclado, monitor, Ethernet, etc). La comunicación se realizó entre una computadora en tierra (e.g. una laptop) y la pico-computadora a bordo del quadrotor, por medio de una red local inalámbrica, a fin de poder modificar el programa después cualquier lugar o punto del planeta. El sistema operativo de la PICO-ITX se almacenó en una memoria USB de tal manera que no sumara mayor peso. El láser utilizado fue el Hokuyo UTM-30LMX (para mayor información, visitar [http://www.hokuyo-aut.jp/02sensor/07scanner/utm\\_30lx.html](http://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html)). Debido a su bajo peso y su rango de trabajo, éste sensor fue efectivo para el control de posición de in micro-vehículo aéreo en zonas interiores.

En la figura 5.1 podemos ver el diagrama de flujo del algoritmo de estimación de pose. La entrada al sistema es el barrido o escaneo actual, el cual es organizado por medio de **contador**. Además, **Escena** y **Modelo** vienen a ser los barridos escena y modelo respectivamente. Los ángulos de Euler vienen dados por **Pitch** y **Roll**. La función **SM(Modelo,Escena)** da como resultado los cambios de orientación y orientación entre dos barridos correspondientes. Finalmente,  $\mathbf{p}_{th}$  y  $\mathbf{r}_{th}$  representan los umbrales máximos permitidos de los ángulos Pitch y Roll para ejecutar el algoritmo de manera adecuada.

### 5.1.2. Controlador de posición

Debido a que nuestro objetivo principal es determinar de manera efectiva la ejecución del algoritmo de estimación de pose y su robustez, se ha considerado un controlador proporcional-derivativo para el control de posición. El componente derivativo del controlador es vital para compensar el hecho de que mantener la posición del quadrotor en un determinado punto genera velocidades inerciales las cuales son compensadas, pero que se podrían incrementar inestablemente

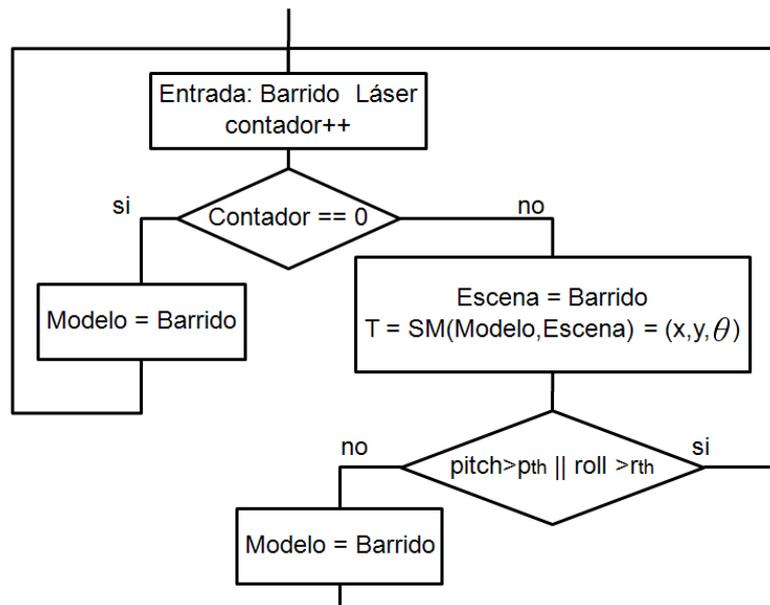


Figura 5.1: Diagrama de flujo del algoritmo de emparejamiento de barrido y su función es representada por  $\mathbf{SM}(\mathbf{Modelo}, \mathbf{Escena})$ . La variable **contador** es usada para guardar el orden en los juegos de datos entrantes.

sin éste componente. En la figura 5.2, vemos el diagrama de bloques del controlador de posición, cuya posición deseada viene dada por  $X_d$ .  $K_p$  y  $K_v$  son sus constantes proporcional y derivativa respectivamente. Estas ganancias fueron determinadas de manera empírica debido a que no se contó con un modelo lineal del helicóptero. El cálculo de la velocidad se hizo mediante el empleo de dos posiciones consecutivas y el tiempo de separación entre las mismas. El tiempo de muestreo se estableció de acuerdo al tiempo de procesamiento del algoritmo de estimación de pose.

## 5.2. Hardware

En ésta sección vamos a detallar el hardware utilizado en el proyecto, el cual puede ser resumido en un quadrotor, un láser, y el sistema de comunicación y procesamiento.

El quadrotor que se muestra en el figura 5.3 tiene una una capacidad de carga

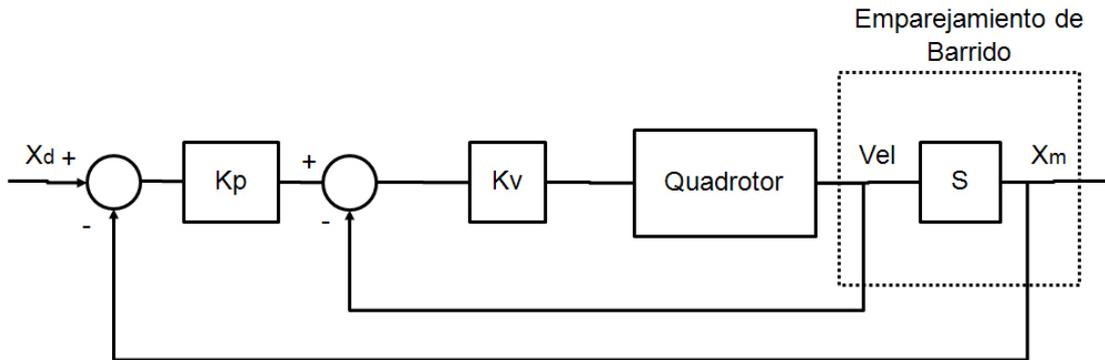


Figura 5.2: Diagrama de bloques del controlador de posición.

de 200 gramos y una potencia de 12 Watts. El controlador de actitud y altitud viene de fábrica (para más información, revisar [http://www.microdrones.com/en\\_home.php](http://www.microdrones.com/en_home.php)), el cual es capaz de estabilizar un ángulo de roll o pitch deseado, así como también una altura de trabajo requerido. La ventaja principal de usar éste dispositivo es que para realizar el control, lo único que hay proporcionar son las señales de control tanto en el eje  $x$  como en el eje  $y$ . Este controlador interno acciona de manera controlada los cuatro motores brushless los cuales son capaces de mover el quadrotor en las tres direcciones traslacionales y las tres direcciones rotacionales.

El sensor Hokuyo UTM-30LMX (ver figura 5.3) es un láser planar, es decir su región de trabajo es en dos dimensiones, el cual cada cierto tiempo (25 ms) almacena un arreglo de puntos correspondientes al contorno del ambiente de trabajo. El voltaje requerido es de 12 Volts y el rango de detección va desde 0.1 hasta los 30 m a una precisión de  $\pm 30$  mm para el rango desde 0.1 hasta 10 m y de  $\pm 50$  mm para el rango desde 10 hasta 30 m. También alcanza un rango de escaneo de  $270^\circ$  a una resolución de  $0.25^\circ$ . Finalmente, su peso es de 370 g.

La computadora utilizada fue la VIA EPIA PX10000G Pico-ITX la cual es idéntica a cualquier computadora personal o de escritorio (para mayor información ver el Apéndice A). Como se mencionó anteriormente



Figura 5.3: Helicóptero quadrotor equipado con un pico-computador y un sensor láser.

ésta computadora soportó eficientemente el sistema operativo Linux bajo la plataforma Linux 7.04. Una memoria USB de 2GB fue la encargada de actuar como disco duro. La comunicación de la computadora con los sensores del Helicóptero se realizó mediante el protocolo RS232 y para la comunicación con la computadora de tierra se empleó una red inalámbrica de Internet.

### 5.3. Experimentos de control

En ésta sección se detallan las consideraciones iniciales de control y los resultados experimentales. Todos los experimentos se realizaron en un ambiente relativamente congestionado de objetos que puedan proporcionar información valiosa para la estimación de posición. El ambiente tiene un área de  $20 \times 15_m$  aproximadamente.

#### 5.3.1. Configuración

La primera consideración fue el hecho de que el sensor láser el cual está ubicado en la parte delantera del quadrotor creaba una momento remanente el cual tenía

que ser compensado de manera experimental. El proceso de compensación es conocido como *Trimming*. Después de unas cuantas pruebas se logró determinar la ganancia offset necesaria para balancear el desequilibrio producido por el peso del láser. La señal de control trabajaría bajo ésta nueva referencia u offset.

Seguidamente se procedió a determinar las ganancias del controlador  $K_p$  y  $K_v$ . Se mantuvo en éstas pruebas una velocidad de traslación medianamente lenta de 0.5 m/s, de tal manera que el operario tuviera el tiempo suficiente de corregir o rescatar al quadrotor en una eventual pérdida de control, lo cual se hizo por medio de un interruptor en el mando de control manual. Además, la velocidad angular yaw fue configurada a cero.

Una forma de poder saber el punto de partida del helicóptero fue mediante una señal en el suelo justo debajo del quadrotor. Esta señal fue una pequeña estrella dorada la cual nos daba una idea de cuan cerca está de conseguir el objetivo de control.

### 5.3.2. Resultados

En el experimento, evaluamos el rendimiento del estimador de posición y el controlador de posición. Para ello, de manera manual, se movió el helicóptero hasta el punto deseado de control que se muestra en la figura 5.4. Una vez posicionado por encima de la estrella dorada, se activo el controlador de posición vía wireless, lo cual considera el instante anterior como la posición deseada ((0,0) m). Lo que sigue a eso son perturbaciones manuales respecto de la posición deseada hasta otro punto en el plano, y es ahí cuando finalmente el controlador es activado para lograr que el quadrotor regrese a su posición de control (0,0) m.

Manteniendo una altura constante y el control manual alerta por si hay un descontrol, se procedió a repetir el procedimiento de dejar que el quadrotor regrese a su posición deseada para después ser perturbado hasta una posición diferente y

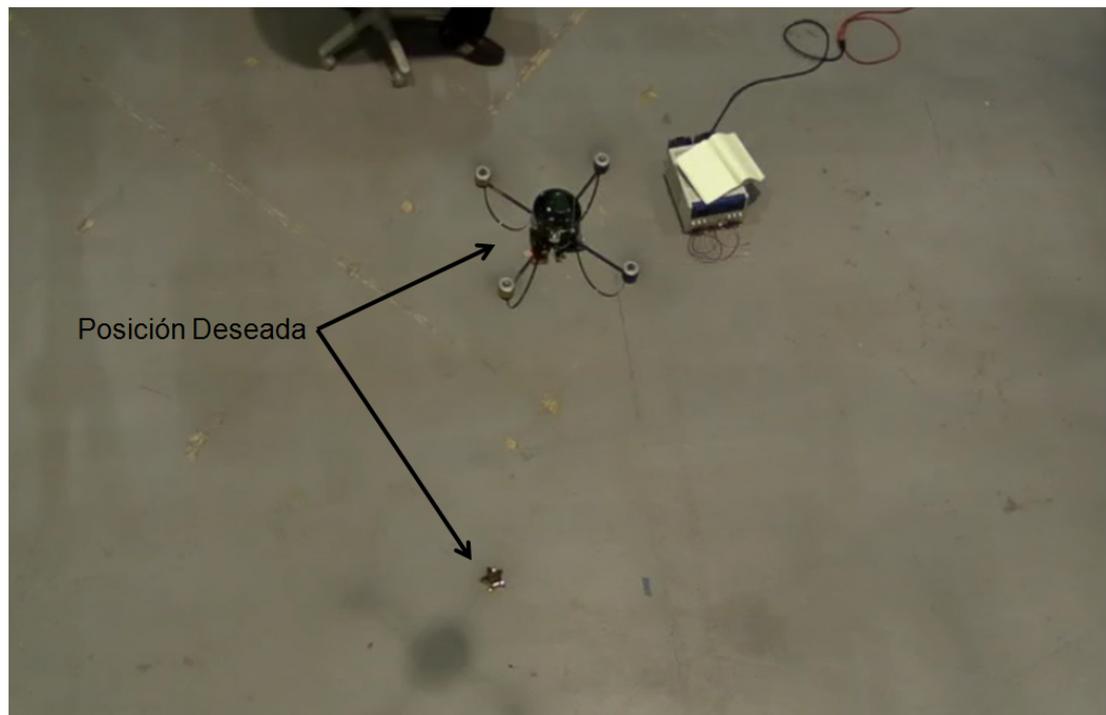


Figura 5.4: Configuración inicial del experimento de control: quadrotor encima de la referencia visual.

dejar nuevamente que retorne a su posición de control. Como podemos apreciar en la figura 5.5, el proceso de perturbar manualmente y activar el controlador se repitió tres veces. En la parte superior del gráfico, las líneas rojas representan cuando el helicóptero operaba en modo manual, las líneas verdes representan el periodo de operación en automático. De ellos podemos concluir que en todo caso de perturbación el controlador fue capaz de restaurar la posición del helicóptero a su posición de control. Las puntas roja y verde de las flechas indican los ejes coordenados  $x$  y  $y$  respectivamente. La saturación a la salida indica el porcentaje de la máxima señal de salida que se empleó durante el experimento de control, en éste caso  $\pm 50\%$ . El  $X_{trim}$  indica la cantidad de offset necesario para compensar el peso desequilibrante del sensor láser. Además el contorno de puntos azules indica la silueta del ambiente de trabajo. Seguidamente en las figuras de la parte inferior podemos apreciar el comportamiento dinámico de la respuesta del sistema en  $x$  y

*y.* Se ve que para ambos casos la respuesta fue como se esperaba, es decir, ambas señales logran mantenerse alrededor del (0,0) m.

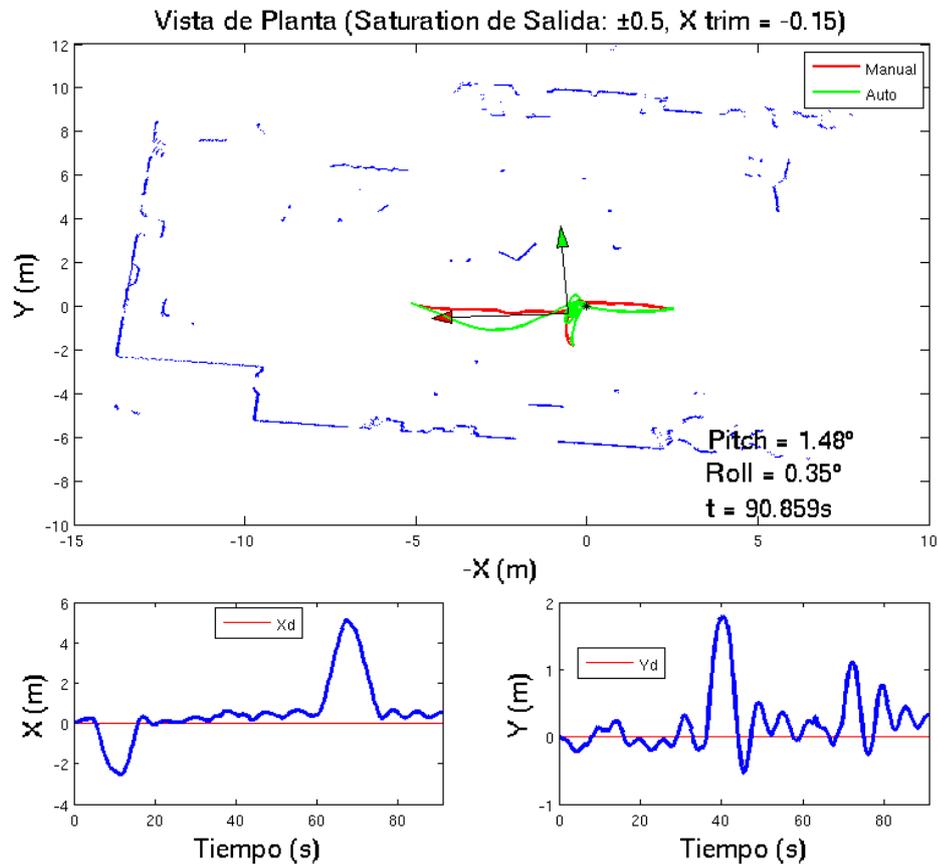


Figura 5.5: Control Hover con perturbaciones manuales.

En la figura 5.6, podemos ver las respuestas dinámicas de las salidas de control, los ángulos pitch y roll y la orientación. En la parte inferior, podemos ver que el ángulo de orientación el cual debe de permanecer fijado a cero grados se mantuvo dentro de un rango permisible de  $\pm 5^\circ$ .

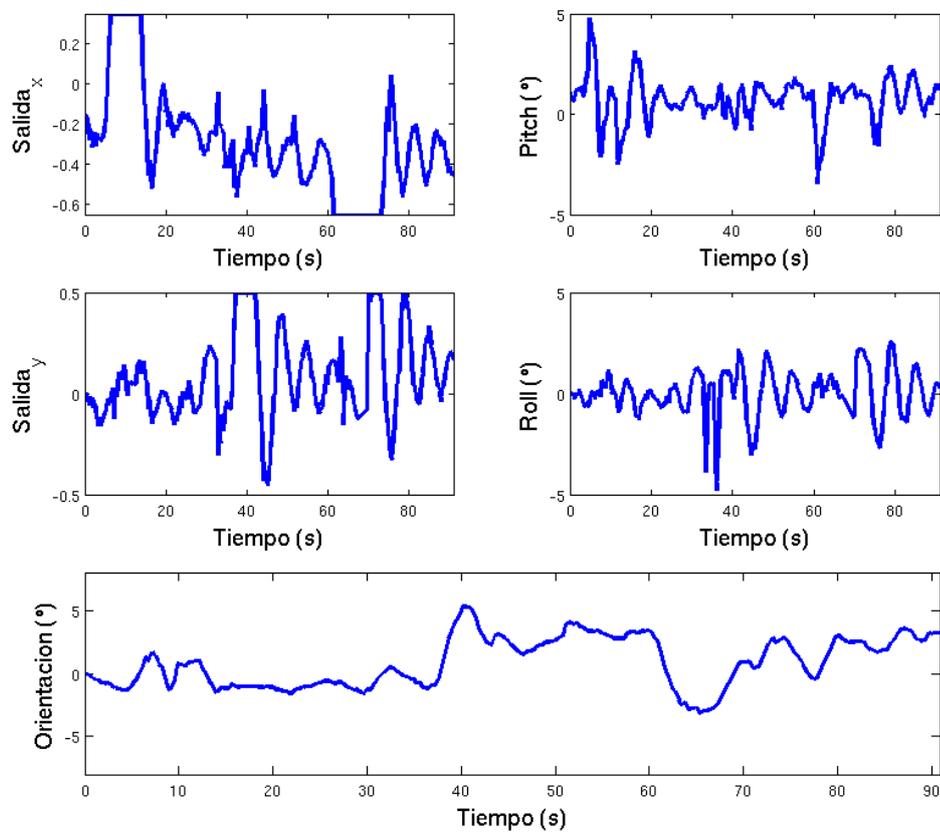


Figura 5.6: Señales y variables de control.

# Conclusiones

En el simulador de vuelo:

1. El modelo no lineal de Gavrillets para un mini helicóptero es altamente fiable, pues considera todos los efectos dinámicos y aerodinámicos. Esto permitió que el simulador de vuelo HeliSi3D pueda representar los efectos más importantes a lo largo del dominio angular de sus variables de estado.
2. El modelo del mini helicóptero resultó ser altamente no lineal, conteniendo inclusive ecuaciones iterativas para la evaluación del empuje. Por tanto, un futuro diseño de control requerirá de la linealización del sistema o un modelo reducido; caso contrario, los métodos de diseño de controladores no lineales serían insuficientes.
3. El periodo de muestreo de simulación depende básicamente de la velocidad de respuesta de los actuadores (servo motores). Por tanto se eligió un tiempo de muestreo cuatro veces más rápido que la velocidad de actualización de entradas en los actuadores equivalente a 5ms.
4. La librería OpenGL en plataforma Visual.net fue apropiada para representar la dinámica de vuelo. No se detectaron problemas durante la animación. Sin embargo, en futuros trabajos sería conveniente añadir otros motores de animación, puesto que el OpenGL, por sí solo, tiene un límite de velocidad en la presentación de los frames o tramas en pantalla.

En el controlador de posición:

1. El algoritmo de la búsqueda en la vecindad fue capaz de reducir el tiempo desde 800 ms hasta 60 ms. Lo cual fue tremendamente útil ya que contábamos con computadora restringida en recursos computacionales.
2. La estimación de movimiento hizo que el algoritmo tuviera un primer tanteo de transformación y de manera efectiva limitó el error de convergencia. de la misma forma el rechazo de barrido basado en la actitud previene de manera apropiada la incorporación de data incorrecta en la estimación de pose. Por tanto, el algoritmo ICP solamente no es capaz de desempeñarse exitosamente en un ambiente cerrado, sino que hace falta de mejoras que ayuden a contrarrestar los problemas de una misión de vuelo real.
3. Se logro controlar el helicóptero en dos dimensiones de manera apropiada usando como estimador de posición el algoritmo de emparejamiento de barrido o scan matching. El controlador basado en éste algoritmo uso en todo momento las estimaciones de posición para calcular el error y así generar las salidas de control, que en todo momento fueron suaves y sin saltos, necesarias para el control de posición.

# Recomendaciones

En el simulador de vuelo:

1. El desarrollo de nuestro algoritmo se hizo en primera instancia en MATLAB, debido a su fácil implementación. Sin embargo, el programa en MATLAB no sólo sirvió para realizar una rápida implementación, sino que también para que sirva de base o referencia del programa hecho en Visual C++. Cualquier definición errónea del tipo de variable en Visual, causaba una divergencia de resultados con respecto a los obtenidos en MATLAB. Por lo tanto, MATLAB fue vital para resolver problemas de implementación de código fuente en C++.
2. Al momento de simular las ecuaciones del sistema, no sólo es importante incluir las restricciones de movimiento contra suelo, sino que también hay que incorporar las restricciones de ángulo (roll y pitch). Por ejemplo, si al momento del despegue en el simulador, el helicóptero lo hace con un ángulo de pitch o roll mayor a  $15^\circ$ , este podría destrozar las palas del rotor principal. Si bien es cierto, hasta ahora sólo se está trabajando a nivel de simulación, pero en la realidad esto podría significar pérdidas cuantiosas. Por tanto es conveniente limitar los ángulos de roll y pitch, especialmente al momento de despegue.
3. Se recomienda utilizar OpenGL como motor de simulación, debido a que es libre y versátil. Existen muchas plataformas de animación, pero

como no son libres, no es posible adecuar nuevos prototipos en esas plataformas. En cambio, OpenGL no sólo dispone de una gran cantidad de documentación, sino que también es fácilmente configurable a nuevos prototipos y rápidamente utilizable en Linux.

En el controlador de posición:

1. Es importante tener en cuenta los distintos algoritmos de estimación de posición disponible en las diferentes revistas científicas, ya que muchos de ellos pueden tener técnicas que pueden solucionar de manera local los problemas en un algoritmo ciertamente grande como el emparejamiento de barrido.
2. Pese al progreso de muchos microprocesadores como los DSP o los FPGA, es conveniente usar una computadora en caso de implementar un controlador, más aun si se trabaja con un vehículo aéreo. Entre las ventajas que se mencionaron anteriormente tenemos que es fácil trabajar con un sistema operativo, también se puede incorporar una red inalámbrica rápidamente y una memoria USB de gran capacidad para hacer la vez de disco duro.
3. Se recomienda mantener siempre un piloto de aeromodelismo experto para tener el control del helicóptero en caso se hagan pruebas de control. Sin él, hay un riesgo enorme de malograr o destruir un aparato que puede llegar a costar US\$ 20 000.

# Bibliografía

- [1] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Autonomous navigation and exploration of a quadrotor helicopter in gps-denied indoor environments. *Association for Unmanned Vehicle Systems International (IARC)*, 2009.
- [2] E. Altug, J. P. Ostrowski, and R. Mahony. Control of a quadrotor using visual feedback. *In Proc. of the IEEE Conf. on Robotics and Automation (ICRA)*, 2002.
- [3] J. L. Blanco. *The Mobile Robot Programming Toolkit (MRPT)*. University of Malaga, [http://babel.isa.uma.es/mrpt/index.php/Main\\_Page](http://babel.isa.uma.es/mrpt/index.php/Main_Page), 2009.
- [4] A. R. S. Bramwell, G. Done, and D. Balmford. *Bramwell's Helicopter Dynamics*. Butterworth-Heinemann, Oxford, second edition, 2001.
- [5] V. Gavrillets. *Autonomous Aerobatic Maneuvering of Miniature Helicopters*. Massachusetts Institute of Technology, Boston, MA, 2003.
- [6] J. Del Cerro Giner. *Arquitectura abierta para el control autónomo y teleoperado de un mini-helicóptero*. Universidad Politécnica de Madrid, Madrid, ES, 2007.
- [7] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. *In Proc. of the IEEE Conf. on Robotics and Automation (ICRA)*, 2009.
- [8] U. B. Hald and M. V. Hesselbæk. *Nonlinear modelling and optimal control of a miniature autonomous helicopter*. Aalborg University, Fredrik Bajers Vej 7 - 9220 Aalborg Ost, Denmark, 2006.
- [9] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin. The stanford testbed of autonomous rotorcraft for multi agent control (starmac). *The 23rd Digital Avionics Systems Conference (DASC)*, 2004.
- [10] T. Jost and H. Hügli. Fast ICP algorithms for shape registration. *Spring-Verlag Berlin Heidelberg*, 2449:91–99, 2002.
- [11] E. Karlsson. *The Kalman Predictor*. <http://www.it.uu.se/edu/course/homepage/systemid/vt09/KalmanPredictor-090215.pdf>, 2009.

- [12] C. Lara, L. Romero, and F. Calderón. A robust iterative closest point algorithm with augmented features. *Spring-Verlag Berlin Heidelberg*, 5317:605–614, 2008.
- [13] R. Mettler. *Identification, Modeling and Characteristics of Miniature Rotorcraft*. Kluwer Academic Publishers, Boston, MA, first edition, 2002.
- [14] MIT Lecture Notes. Lecture 2. *16.61 Aerospace Dynamics*, 1:16–21, 2003.
- [15] G. D. Padfield. *Helicopter Flight Dynamics: The theory and Application of Flying Qualities and Simulation Modeling*. AIAA Education Series, Reston, VA, second edition, 2007.
- [16] Raul Rojas. *The Kalman Filter*. <http://robocup.mi.fu-berlin.de/buch/kalman.pdf>, 2009.
- [17] J. Seddon. *Basic Helicopter Aerodynamics*. Oxford, first edition, 1990.
- [18] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley and sons, INC, Hoboken, NJ, second edition, 2003.
- [19] A. Tayebi and S. McGilvray. Attitude stabilization of a four-rotor aerial robot. *43rd IEEE Conference on Decision and Control (CDC)*, 2004.
- [20] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Boston, USA, 2005.
- [21] K. P. Valavanis. *Advances in unmanned aerial vehicles*, volume 33. Springer, Tampa, FL, 2007.
- [22] C. M. Velez. Diseño, implementación y prueba de un sistema de control y navegación para un mini-helicóptero robot-colibrí. *Grupo de Investigación en Sistemas de Control Digital*, 1:1–30, 2006.
- [23] W. R. Young. *Los Helicópteros I*, volume 27. Time-Life, Folio, Muntaner, Barcelona, 1982.

## Apéndice A

### Especificaciones técnicas del pico-computador VIA EPIA PX10000G Pico-ITX

## Pico-ITX Series

# VIA EPIA PX

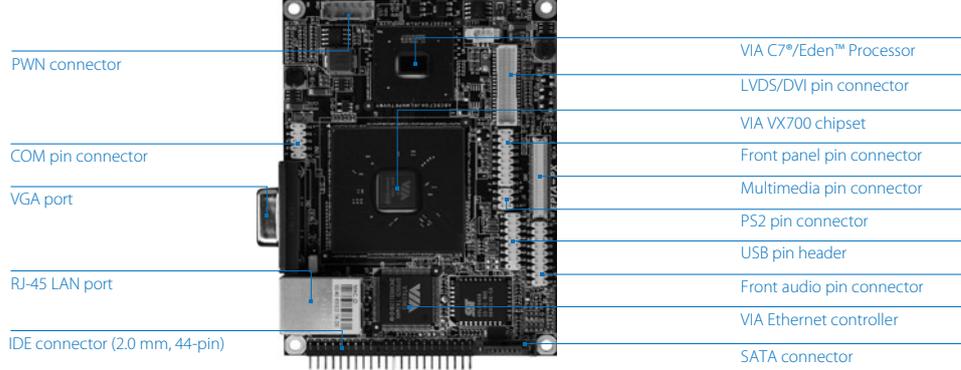
Pico-ITX C7<sup>®</sup>/Eden™ ULV  
Board with MPEG-2 &  
4, WMV9 and LVDS/DVI  
Support

## Features

- Integrated VIA UniChrome™ Pro II 3D/2D AGP graphics with MPEG-2/4 and WMV9 decoding acceleration
- Supports DDR2 533/667 SDRAM (SODIMM)
- Audio connector for Line-in, Mic-in, S/PDIF in, and 5.1 channels audio output
- Low profile design: All back panel I/O in pin header format except for the RJ-45 LAN and VGA port



## Board Placement



## Specifications

Model Name	EPIA PX10000G	EPIA PX5000EG
<b>Processor</b>	1.0GHz VIA C7 <sup>®</sup>	500MHz VIA Eden™ ULV
<b>Chipset</b>	VIA VX700 Unified Digital Media IGP chipset	
<b>System Memory</b>	1 x DDR2 533/667 SODIMM socket (effective speed to 533MHz) Up to 1GB memory size	
<b>VGA</b>	Integrated VIA UniChrome™ Pro II 3D/2D AGP graphics with MPEG-2/4 and WMV9 decoding acceleration	
<b>Onboard IDE</b>	1 x UltraDMA 133/100/66 connector (2.0 mm 44-pin header)	
<b>Onboard Serial ATA</b>	1 x SATA connector	
<b>Onboard LAN</b>	1 x VIA VT6106S 10/100 Mbps Ethernet controller	
<b>Onboard Audio</b>	VIA VT1708A High Definition Audio Codec	
<b>Onboard I/O Connectors</b>	1 x USB connector for 4 USB 2.0 ports 1 x COM port connector 1 x PS2 mouse/keyboard connector 1 x LVDS/DVI pin connector, for 18/24-bit dual channel LVDS panel (if without using DVI) 1 x Multimedia connector to support external TV-out interface, video capture port interface and low pin count interface (an add-on card is required) 1 x Audio connector for Line-in, Mic-in, S/PDIF in, and 5.1 channels audio output 1 x Fan connector for CPU fan 1 x Pico-ITX power connector	
<b>Back Panel I/O</b>	1 x RJ-45 LAN port 1 x VGA port	
<b>BIOS</b>	Award BIOS 4/8Mbit flash memory	
<b>Operating System</b>	Windows XP, Linux, Win CE, XPe	
<b>System Monitoring Management</b>	CPU temperature reading (manufacturing option) Watch Dog Timer (manufacturing option) Keyboard Power-on, Time Power-on System power management AC power failure recovery	
<b>Operating Temperature</b>	0°C ~ 50°C	
<b>Operating Humidity</b>	0% ~ 95% (relative humidity; non-condensing)	
<b>Form Factor</b>	Pico-ITX (10-layer) 10 cm x 7.2 cm	