

CIS1330NG01

Seguimiento y realidad aumentada con un dron teleoperado

Mauricio Armando González Rodríguez

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.
NOVIEMBRE 19 DE 2013

CIS1330NG01

Seguimiento y realidad aumentada con un dron teleoperado

Autor:

Mauricio Armando González Rodríguez

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE
LOS REQUISITOS PARA OPTAR AL TITULO DE INGENIERO DE SISTEMAS

Director

Pedro Fabián Cárdenas Herrera Ph.D.

Jurados del Trabajo de Grado

Ricardo Emiro Ramírez Heredia Ph.D.

Leonardo Flórez Valencia Ph.D.

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~CIS1330NG01/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA DE SISTEMAS
BOGOTÁ, D.C.
NOVIEMBRE 19 DE 2013

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

Rector Magnífico

Joaquín Emilio Sánchez García S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Decano del Medio Universitario Facultad de Ingeniería

Padre Sergio Bernal Restrepo S.J.

Director de la Carrera de Ingeniería de Sistemas

Ingeniero Germán Alberto Chavarro Flórez

Director Departamento de Ingeniería de Sistemas

Ingeniero Rafael Andrés González Rivera

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Υ

*A mi querido Padre celestial
de la luz verdadera de los hombres piadosos y las mujeres piadosas,
a mi querida Madre celestial
de la luz verdadera de los hombres piadosos y las mujeres piadosas,
a la Santísima Trinidad:
DIOS PADRE, HIJO Y ESPIRITU SANTO,
a la Divina Providencia,
a la Sagrada Familia:
Jesús, José y María,
a los tres arcángeles:
San Miguel, San Gabriel y San Rafael,
a los santos ángeles de la guarda,
a la comunión de los santos,
a las benditas almas del purgatorio,
a la iglesia militante,
a mis padres y a Luisa Fernanda.*

*Con amor, gratitud y respeto a la Santa Cruz
y
generosa oración del Santo Rosario,
para que la llaga de amor hiera todas las almas.*

†

Contenido

INTRODUCCIÓN	14
I - DESCRIPCION GENERAL DEL TRABAJO DE GRADO	15
1. OPORTUNIDAD, PROBLEMÁTICA, ANTECEDENTES	15
1.1 Descripción del contexto.....	15
1.2 Formulación del problema que se resolvió.....	18
1.3 Justificación.....	18
1.4 Impacto Esperado.....	18
2. DESCRIPCIÓN DEL PROYECTO	18
2.1 Visión global.....	18
2.2 Objetivo general.....	18
2.3 Fases Metodológicas y conjunto de objetivos específicos.....	18
2.4 Método que se propuso para satisfacer cada fase metodológica.....	19
2.4.1 Fase del ciclo de diseño.....	20
2.4.2 Fase de producción.....	21
II - MARCO TEÓRICO	22
1. MARCO CONTEXTUAL	22
2. MARCO CONCEPTUAL	22
2.1 Control remoto Wii™.....	22
2.2 APIs para el control remoto Wii.....	24
2.3 Unmanned Aircraft (UA).....	24
2.4 Unmanned Aircraft System (UAS).....	25
2.5 Retos éticos y legales de los drones.....	26
2.6 Parrot AR.Drone 1.0.....	30
2.7 Movimientos básicos del Parrot AR.Drone 1.0.....	32
2.8 Comando de control de movimiento del Parrot AR.Drone 1.0.....	36
2.9 Cámaras del Parrot AR.Drone 1.0.....	37
2.10 Giroscopios, acelerómetro y altímetro del Parrot AR.Drone 1.0.....	37
2.11 APIs para el Parrot AR.Drone 1.0.....	38
2.12 DFRobotShop Rover V2 (Xbee® Kit).....	38
2.13 API para el DFRobotShop Rover V2 (Xbee® Kit).....	39
2.14 Teleoperación.....	39
2.15 Augmented Reality (AR).....	39
2.16 Marcadores visuales.....	40
2.17 APIs para la AR.....	41
2.18 Controlador PID.....	42
2.19 Pseudocódigo para implementar un controlador PID.....	43
2.20 Filtro Paso Bajo de primer orden.....	43
2.21 Pseudocódigo para implementar un Filtro Paso Bajo de primer orden.....	44

III – DESARROLLO DEL TRABAJO	45
1. FASE DEL CICLO DE DISEÑO	45
1.1 Selección de sensores.....	45
1.2 Selección de librerías.....	45
1.3 Selección de estrategia.....	45
1.4 Selección del algoritmo de detección, seguimiento y realidad aumentada.....	46
1.5 Definición del modelo para controlar el dron.....	46
1.6 Proyección de los estados del prototipo.....	46
1.7 Proyección de la clase FiltroPasoBajo.....	47
1.8 Proyección de la clase ControladorPID.....	47
1.9 Pseudocódigo del algoritmo de detección, seguimiento y realidad aumentada.....	48
2. FASE DE PRODUCCIÓN	50
2.1 Descripción del montaje.....	50
2.2 Instalación de las librerías de Processing.....	50
2.3 Prueba inicial de funcionamiento.....	53
2.4 Implementación del prototipo.....	53
2.5 Sorpresas durante la implementación del prototipo.....	56
2.6 Prueba del prototipo.....	56
2.7 Comunicación de resultados.....	58
2.8 ¿ Cómo se usa el prototipo ?.....	58
IV - RESULTADOS Y REFLEXIÓN SOBRE LOS MISMOS	62
1. CUMPLIMIENTO DEL OBJETIVO GENERAL	62
2. CUMPLIMIENTO DE LOS OBJETIVOS ESPECÍFICOS	62
3. APORTE A LA SOLUCIÓN DEL PROBLEMA	63
4. IMPACTO PRESENTE Y FUTURO	63
5. MEJORAS DEL PROTOTIPO	64
V – CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS	65
1.CONCLUSIONES	65
2.RECOMENDACIONES	65
3.TRABAJOS FUTUROS	65
VI - REFERENCIAS	66
VII - ANEXOS	72
ANEXO 1. GLOSARIO	72
ANEXO 2. POST-MORTEM	73
1. Metodología propuesta vs. Metodología realmente utilizada.....	73

2. <i>Actividades propuestas vs. Actividades realizadas</i>	73
3. <i>Efectividad en la estimación de tiempos del proyecto</i>	73
4. <i>Costo estimado vs. Costo real del proyecto</i>	75
5. <i>Efectividad en la estimación y mitigación de los riesgos del proyecto</i>	76
ANEXO 3. MARCAS USADAS EN ESTE TRABAJO	78

Índice de ilustraciones

Ilustración 1.Control remoto Wii™.....	15
Ilustración 2.Parrot AR.Drone 1.0.....	17
Ilustración 3.DFRobotShop Rover V2 (Xbee® Kit) Fuente: RobotShop Distribution Inc.....	17
Ilustración 4.Fase del ciclo de diseño y fase de producción del modelo unificado de diseño propuesto por Karl T. Ulrich [44].....	19
Ilustración 5.Pasos de la fase del ciclo de diseño propuestos por Karl T. Ulrich en su modelo [44].....	20
Ilustración 6.Elementos de un UAS [6].....	26
Ilustración 7.Parrot AR.Drone 1.0 con su casco protector para vuelos en espacios interiores.	30
Ilustración 8.Fuerza de empuje.....	33
Ilustración 9.Ascenso.....	33
Ilustración 10.Descenso.....	33
Ilustración11.Movimiento 1.....	34
Ilustración 12.Movimiento 2.....	34
Ilustración 13.Movimiento 3.....	34
Ilustración 14.Movimiento 4.....	35
Ilustración 15.Rotación 1.....	35
Ilustración 16.Rotación 2.....	35
Ilustración 17.Alabeo.....	36
Ilustración 18.Cabeceo.....	36
Ilustración 19.Guiñada.....	37
Ilustración 20.Continuo realidad-virtualidad de Milgram [21].....	40
Ilustración 21.Marcador ARToolKit.....	41
Ilustración 22.Diagrama de estados del prototipo.....	47

Ilustración 23. Diagrama de la clase FiltroPasoBajo.....	47
Ilustración 24. Diagrama de la clase ControladorPID.....	48
Ilustración 25. Rover con marcador visual ARToolKit de 8cmX8cm.....	54
Ilustración 26. Rover con marcador visual ARToolKit tamaño carta.....	54
Ilustración 27. Cuadrícula pintada sobre el piso del cuarto de vuelo.....	57
Ilustración 28. Fotograma del vídeo de la caja negra del dron.....	58
Ilustración 29. Imagen inicial del prototipo.....	59
Ilustración 30. Ícono de seguimiento debajo del ícono de carga de la batería.....	60
Ilustración 31. Esfera amarilla sobre el marcador visual ARToolKit.....	60
Ilustración 32. Diagrama PERT/CPM propuesto.....	73
Ilustración 33. Diagrama PERT/CPM ajustado al principio del semestre.....	74

Índice de tablas

Tabla 1.Comparación de algunos cuadricópteros similares.....	32
Tabla 2.Estados de ejecución de las fases y sus objetivos específicos.....	63
Tabla 3.Cronograma detallado con las horas de trabajo diario propuesto.....	74
Tabla 4.Cronograma detallado con las horas de trabajo diario ajustado al principio del semestre.....	75
Tabla 5.Costo estimado del trabajo de grado.....	76
Tabla 6.Costo real del trabajo de grado.....	76
Tabla 7.Riesgos estimados para el trabajo de grado.....	77

ABSTRACT

In this writing it was developed a software prototype that enables a Parrot AR.Drone 1.0 quadrotor to track a DFRobotShop Rover V2 and add augmented reality. The approach used requires an ARToolKit visual marker placed on the rover and uses the camera built into the drone, a 1st order Low Pass Filter to filter drone's altimeter noise and some PID controllers to generate the movements of pitch, roll and vertical speed to align the center of the drone's camera image with the center of the rover's ARToolKit visual marker.

RESUMEN

En este escrito se desarrolló un prototipo de software que permite a un cuadrirotor Parrot AR.Drone 1.0 seguir a un DFRobotShop Rover V2 y agregar realidad aumentada. El enfoque que se empleó requiere un marcador visual ARToolKit colocado sobre el rover y usa la cámara incorporada en el dron, un Filtro Paso Bajo de primer orden para filtrar el ruido del altímetro del dron y unos controladores PID para generar los movimientos de cabeceo, alabeo y velocidad vertical que permiten alinear el centro de la imagen de la cámara del dron con el centro del marcador visual ARToolKit del rover.

RESUMEN EJECUTIVO

En este trabajo de grado se desarrolló un prototipo de software que permite a un cuadricóptero Parrot AR.Drone 1.0 detectar y seguir automáticamente un vehículo terrestre no tripulado de tipo DFRobotShop Rover V2 y agregar realidad aumentada a las imágenes percibidas por el dron desde arriba. La solución que se encontró no usa un marco de referencia absoluto, sino que manipula la imagen de la cámara incorporada en el dron para determinar la posición relativa del uno con respecto al otro y requiere el uso de un marcador visual ARToolKit colocado sobre el rover, un Filtro Paso Bajo de primer orden para filtrar el ruido del altímetro del dron y unos controladores PID para generar los movimientos de cabeceo, alabeo y velocidad vertical que permiten alinear el centro de la imagen de la cámara del dron con el centro del marcador visual ARToolKit del rover.

Usando el método de ensayo y error se determinaron los siguientes valores para que el dron pueda seguir al rover usando los parámetros ajustables de los controladores PID. Para el cabeceo $K_p=0.001$, $K_i=0.00001$ y $K_d=0.03$,para el alabeo: $K_p=0.001$, $K_i=0.00001$ y $K_d=0.03$. Para la velocidad vertical $K_p=0.001$, $K_i=0.000001$ y $K_d=0.001$. El ruido del altímetro del dron se manejó con un Filtro Paso Bajo de primer orden con un $\alpha=0.9$.

Con estos valores se logró mantener el dron a una altura aproximada de 1 metro siguiendo al rover a una distancia de hasta 30 centímetros alrededor del mismo.

INTRODUCCIÓN

Este trabajo de grado es un modesto primer acercamiento que realiza el autor a la robótica móvil. No buscó otra cosa diferente a la de disfrutar los robots que se tuvieron que comprar con la excusa de terminar una carrera. No codició el resultado final porque lo más importante fue divertirse con un dron, un rover, un control remoto y una computadora portátil.

En el capítulo I se encuentra una descripción general del trabajo de grado, en el capítulo II el marco teórico que estructura y da solidez al mismo, en el capítulo III el desarrollo del trabajo realizado en dos fases, en el IV los resultados obtenidos y en el V las conclusiones del mismo.

I - DESCRIPCION GENERAL DEL TRABAJO DE GRADO

1. Oportunidad, Problemática, Antecedentes

1.1 Descripción del contexto

El control remoto Wii™ es un dispositivo de entrada y salida tangible, portátil y de bajo costo que se puede usar para teleoperar un micro-Unmanned Aircraft System (micro-UAS).

Como dispositivo de entrada de datos percibe la aceleración con respecto a tres ejes, detecta emisiones de luz infrarroja y permite pulsar sus botones. Como dispositivo de salida de datos suministra al usuario realimentación táctil, visual y auditiva [27]. El control remoto se comunica con la computadora usando Bluetooth® y usa como fuente de energía dos baterías AA. Adicionalmente ofrece diversos accesorios que amplían su forma y agregan nuevos tipos de interacción. En la Ilustración 1 se puede observar un control remoto Wii™.



Ilustración 1.Control remoto Wii™.

Processing es un lenguaje de programación creado en el año 2001 por Ben Fry y Casey Reas cuando ambos eran estudiantes del Grupo de Estética y Computación del Laboratorio de

Medios del Instituto Tecnológico de Massachusetts. Fue construido para hacer más fácil la programación de gráficas interactivas y cuenta con más de cien librerías desarrolladas por su comunidad que lo enriquecen con asombrosas aplicaciones [37].

El cuadricóptero Parrot AR.Drone 1.0 es un juguete volador impulsado por cuatro rotores y se puede considerar un micro-UAS o micro-Unmanned Aerial Vehicle (micro-UAV) de tipo Vertical Take Off and Landing (VTOL). Fue diseñado para sostener juegos de realidad aumentada y ser teleoperado con teléfonos inteligentes iPhone® y Adroid™ y otros dispositivos tales como: iPads®, iPods touch®, tabletas Android™ y computadoras personales [35]. Hoy se usa como una plataforma para la educación y la investigación robótica [8][12][18][19][25][29][38][39][43]. El aparato tiene una tarjeta principal y otra dedicada exclusivamente a la navegación. La tarjeta principal administra los cuatro rotores, el vídeo de las dos cámaras incorporadas (una que mira al frente y otra que mira hacia abajo), la conexión Wi-Fi®, la conexión USB, el despegue, el aterrizaje y la estabilización de la nave. La tarjeta de navegación detecta la altura, la aceleración y las rotaciones del vehículo aéreo. La fusión de los datos, que proporcionan los sensores de esta tarjeta, se logra mediante el uso de filtros digitales [12] que permiten calcular los ángulos de Euler necesarios para la estabilización de la nave. El cuadrirotor puede alcanzar velocidades superiores a 5 m/s y su autonomía está en el rango de 10 a 15 minutos de vuelo continuo, por carga de batería [12]. La tarjeta principal corre el sistema operativo BusyBox Linux® [25]. En la Ilustración 2 se puede apreciar un Parrot AR.Drone 1.0.

El robot oruga DFRobotShop Rover V2 (Xbee® Kit) es un robot móvil, con forma de tanque, compatible con el microcontrolador Arduino Uno USB Rev 3. En la Ilustración 3 se puede atisbar un DFRobotShop Rover V2 (Xbee® Kit).

El reto consistió en descubrir las condiciones en las que puede ser teleoperado un cuadricóptero Parrot AR.Drone 1.0 dedicado a labores de detección y seguimiento automático de un DFRobotShop Rover V2, administrando todos los recursos en Processing y comunicando al usuario las distintas alternativas de interacción aumentando la realidad percibida.



Ilustración 2. Parrot AR.Drone 1.0.

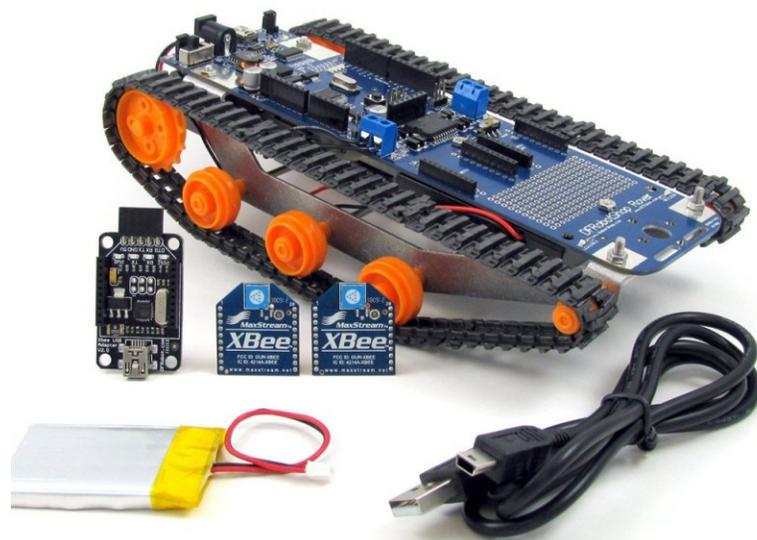


Ilustración 3. DF RobotShop Rover V2 (Xbee® Kit) Fuente: RobotShop Distribution Inc.

1.2 Formulación del problema que se resolvió

¿ Cómo teleoperar un cuadricóptero Parrot AR.Drone 1.0 en tareas de detección y seguimiento automático de un DFRobotShop Rover V2 empleando realidad aumentada ?

1.3 Justificación

Este trabajo de grado apoyó el desarrollo de una plataforma de bajo costo que puede ser usada para el control de robots aéreos dedicados a tareas de detección y seguimiento automático de objetos.

1.4 Impacto Esperado

Este trabajo de grado contribuyó a la comunidad de la robótica aérea que cuenta con pequeños presupuestos para adquirir hardware y software para sus proyectos.

2. Descripción del Proyecto

2.1 Visión global

Se implementó un prototipo de software que permite a un cuadricóptero Parrot AR.Drone 1.0 seguir a un DFRobotShop Rover V2 y añadir realidad aumentada. El enfoque que se empleó requiere un marcador visual ARToolKit colocado sobre el rover y usa la cámara incorporada en el dron, un Filtro Paso Bajo de primer orden para filtrar el ruido del altímetro del dron y unos controladores PID para generar los movimientos de cabeceo, alabeo y velocidad vertical que permiten alinear el centro de la imagen de la cámara del dron con el centro del marcador visual ARToolKit del rover.

2.2 Objetivo general

El objetivo general fue desarrollar un prototipo de software que permitiera teleoperar un cuadricóptero Parrot AR.Drone 1.0 en tareas de detección y seguimiento automático de un DFRobotShop Rover V2 previamente etiquetado con un marcador visual ARToolKit en un área de vuelo libre de obstáculos, ubicada en un espacio interior con buena iluminación.

2.3 Fases Metodológicas y conjunto de objetivos específicos

Fase del ciclo de diseño:

El objetivo específico de esta fase fue seleccionar el algoritmo que se empleó para que el robot aéreo detectara, siguiera y mostrara la información del objeto marcado usando realidad aumentada.

Fase de producción:

El objetivo específico de esta fase fue implementar el prototipo de software con el algoritmo elegido y considerando el uso de marcadores visuales ARToolKit para demostrar su funcionamiento.

2.4 Método que se propuso para satisfacer cada fase metodológica

Este trabajo de grado usó como metodología el **modelo unificado de diseño**, propuesto por Karl T. Ulrich que pretende unir la actividad del diseño en todos los dominios y aportar un marco de referencia para la creación de artefactos en la sociedad. En este contexto se considera que la actividad del diseño es: “concebir y dar forma a artefactos que resuelven problemas” y un artefacto es: “cualquier producto de creación intencional incluyendo bienes físicos, servicios, software, gráficos, construcciones, paisajes, organizaciones y procesos” [44].

Este modelo tiene sólo dos fases: la fase del ciclo de diseño y la fase de producción como se muestra en la Ilustración 4.

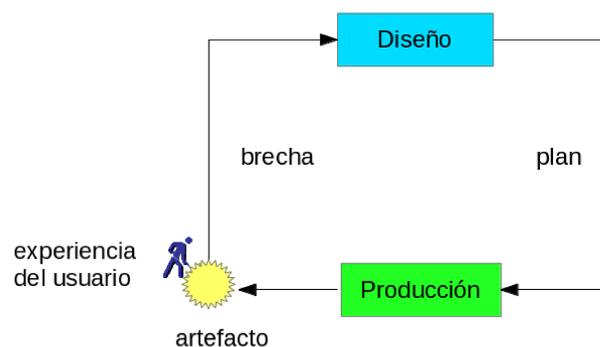


Ilustración 4. Fase del ciclo de diseño y fase de producción del modelo unificado de diseño propuesto por Karl T. Ulrich [44].

Todo comienza porque se percibe un problema que hay que resolver, existe una brecha que hay que superar, un usuario experimenta una insatisfacción y la idea es llevar a ese usuario a la plena satisfacción. Para poder complacer al usuario es necesario hacer un plan para crear un artefacto que resuelve el problema. Una vez se cuenta con un plan concreto se puede producir ese nuevo artefacto. La fase del ciclo de diseño “transforma una brecha en un plan” y la fase de producción “transforma un plan en un artefacto” [44].

2.4.1 Fase del ciclo de diseño

Consiste en percibir la brecha, definir el problema, explorar las alternativas y seleccionar el plan [44] como se revela en la Ilustración 5.

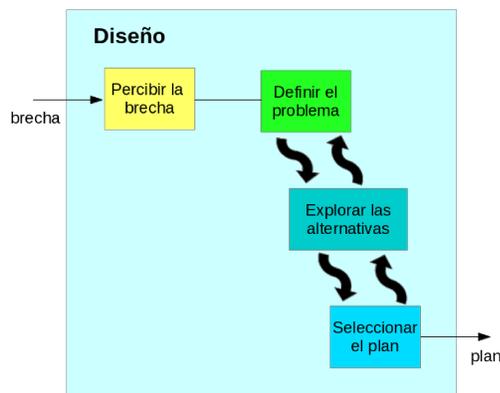


Ilustración 5. Pasos de la fase del ciclo de diseño propuestos por Karl T. Ulrich en su modelo [44].

Cuando el primer artefacto producido no resuelve el problema percibido el ciclo total de diseño se itera múltiples veces [44].

Las actividades desarrolladas durante esta fase fueron las siguientes:

- Se seleccionaron los sensores más apropiados del control remoto Wii™ y del Parrot AR.Drone 1.0 para usarlos adecuadamente en el entorno específico de Processing.

- Se seleccionaron las librerías desarrolladas en Processing que permitieron conectar el control remoto Wii™ vía Bluetooth® y el Parrot AR.Drone 1.0 vía Wi-Fi®.
- Se seleccionaron las librerías desarrolladas en Processing que facilitaron el uso de la Realidad Aumentada.
- Se seleccionó la estrategia de detección y seguimiento del objeto marcado y la distribución de la visión aumentada que se mostró por pantalla.

2.4.2 Fase de producción

Consiste en transformar el plan en un artefacto [44].

Las actividades que se desarrollaron durante esta fase fueron:

- Se construyó el prototipo de software implementando las comunicaciones (portátil, control remoto Wii™, Parrot AR.Drone 1.0) y el algoritmo seleccionado para realizar la detección, el seguimiento y la realidad aumentada.
- Se probó el prototipo con el DFRobotShop Rover V2 marcado en el área de vuelo.
- Se comunicaron los resultados del prototipo en la memoria de trabajo de grado y en la página web.

II - MARCO TEÓRICO

1. Marco Contextual

El beneficiario de este trabajo de grado es la comunidad de la robótica aérea que cuenta con pequeños presupuestos para realizar investigaciones y desarrollos. Desde el punto de vista del hardware se utilizaron elementos de bajo costo. Desde el punto de vista del software se emplearon librerías de código abierto que facilitaron el uso, cambio, mejoramiento y redistribución del software.

La aplicación de este trabajo de grado es para la industria de la robótica aérea porque se aportó un controlador que permite realizar el seguimiento automático de un objeto previamente etiquetado. El controlador se puede ajustar a otros modelos de drones.

2. Marco Conceptual

2.1 Control remoto Wii™

Una **interfaz de usuario tridimensional** es aquella que “implica la interacción humano computadora donde las tareas del usuario son llevadas a cabo en un contexto espacial tridimensional usando dispositivos de entrada de tres dimensiones, o de dos dimensiones, con asignaciones directas a tres dimensiones. En otras palabras, involucra dispositivos de entrada y técnicas de interacción para controlar efectivamente contenido tridimensional, altamente dinámico, generado por computadora” [26].

Otro de los conceptos fundamentales que ilustran esta sección es el de los dispositivos espacialmente convenientes para interfaces de usuario tridimensional.

“Los **dispositivos espacialmente convenientes** implican tres aspectos importantes:

- Datos espaciales. El dispositivo proporciona datos de entrada tridimensional aunque estos sean incompletos, propensos a error o condicionales.
- Funcionalidad. El dispositivo contiene una serie de útiles sensores, emisores e implementos de interfaz.

- Diseño de producto genérico. El dispositivo es barato, duradero, fácilmente configurable y robusto” [46].

Mientras que los dispositivos tradicionales de entrada tridimensional tales como el Polhemus Fastrak®, el InterSense IS-900 y el Vicon cuestan miles de dólares, los dispositivos espacialmente convenientes cuestan pocos dólares. Por ejemplo el 3Dconnexion SpaceNavigator™ cuesta 99 dólares, el nuevo modelo del control remoto Wii™ cuesta 32 dólares y cuenta con clones de más bajo costo tales como el Nyko Wii Wand+, el Mad Catz Wii Remote Plus y el Sunflex Snakebyte Premium Remote XL+.

El 3Dconnexion SpaceNavigator™ tiene seis grados de libertad porque puede combinar tres translaciones (adelante-atrás, arriba-abajo, izquierda-derecha) con tres rotaciones (cabeceo, alabeo, guiñada) sobre cada uno de los tres ejes perpendiculares X,Y,Z.

El control remoto Wii™ no tiene seis grados de libertad sino que maneja la aceleración y la percepción óptica intermitente de emisiones infrarrojas [46].

En la versión antigua del control remoto Wii™ (Wii Remote) se podía agregar un accesorio conocido con el nombre de Wii Motion Plus que era un giroscopio que agregaba el cambio de la orientación en cada uno de los tres ejes perpendiculares X, Y, Z o aportaba también la velocidad angular [46]. En la nueva versión del control remoto Wii™ (Wii Remote Plus) este accesorio ya viene integrado. Es interesante anotar que el clon Nyko Wii Wand+ fue el primero en hacer esta integración adelantándose a Nintendo®. Hoy todos los clones ofrecen una versión integrada.

Debido a sus características técnicas, el control remoto Wii™ es un dispositivo que no puede entregar datos espaciales correspondientes a una posición exacta del mundo real [46].

En el campo de las interfaces de usuario, el control remoto Wii™ responde a la tendencia del retorno a los dispositivos y controles físicos pronosticada por el diseñador Donald A. Norman. “El retorno a dispositivos físicos, donde controlamos las cosas por medio del movimiento físico del cuerpo al girar, mover y manipular dispositivos mecánicos

apropiados.” “El retorno a controles mecánicos conectados con inteligentes procesadores implantados y comunicaciones” [32].

Este retorno a la acción física, más que a la virtual, hace que algunas personas olviden la diferencia entre el dispositivo mecatrónico real y su percepción visual. Es famoso el caso de varios usuarios que cuando juegan bolos en su Nintendo® Wii™, en medio de la excitación del juego, sueltan el control remoto como si este fuera una bola de bolos real que termina en moñona contra la pantalla del televisor [33].

En el campo del diseño de la interacción existen preguntas abiertas y sin resolver como por ejemplo: ¿Cómo se pueden incorporar gestos en los menús? ¿Cómo se puede informar a una persona que usa una aplicación qué alternativas tiene? [34].

Hasta donde se sabe se han investigado algunas técnicas usando el control remoto Wii™ como apuntador en menús visuales de dos dimensiones [14] y menús auditivos esféricos de 360 grados que usan sonido tridimensional y no requieren realimentación visual [23] , pero las preguntas siguen aún sin respuesta.

2.2 APIs para el control remoto Wii

Gracias a los esfuerzos de muchas personas que han realizado ingeniería inversa a este dispositivo se cuenta con una gran variedad de APIs desarrolladas en diferentes lenguajes de programación. En C# y Visual Basic (Managed Library for Nintendo's Wiimote), en C++ (WiiYourself!, WiiuseC++), en C (Cwiid, Wiiuse), en Python(PyWii), en Java (WiiRemoteJ, motej, WiiuseJ), en Processing (wrj4p5, <http://sourceforge.jp/projects/wrj4p5/>).

En Japón, ClassicIII creó la librería wrj4p5 y la liberó bajo Licencia Pública General Reducida de GNU.

2.3 Unmanned Aircraft (UA)

“Es un dispositivo usado o que pretende ser usado para realizar un vuelo en el aire que no tiene piloto a bordo. Esta invención excluye misiles, armas y ojivas explosivas, pero incluye todas las clases de aviones, helicópteros, dirigibles y aeronaves tipo powered-lift, sin llevar un piloto a bordo. En la categoría UA no se incluyen globos tradicionales, cohetes, aeronaves atadas y planeadores sin motor” [20].

2.4 Unmanned Aircraft System (UAS)

“Es un UA y sus elementos asociados relacionados con operaciones seguras, que puede incluir estaciones de control (en tierra, barco o aire), enlaces de control, equipo de soporte, cargas útiles, sistemas de terminación de vuelo y equipo de lanzamiento y recuperación” [20].

“En los tiempos modernos una aeronave no tripulada ha llegado a significar un vehículo aéreo autónomo o pilotado remotamente, que vuela imitando las maniobras de una nave tripulada o pilotada por un ser humano. Incluso el nombre asignado a una nave desocupada ha cambiado a través de los años de acuerdo a como lo han considerado los fabricantes de aeronaves, las autoridades de la aviación civil y los militares. Torpedo aéreo, radio controlado, remotamente pilotado, a control remoto, de control autónomo, vehículo sin piloto, Unmanned Aerial Vehicle (UAV) y dron (del inglés drone) son sólo algunos de los nombres usados para describir una máquina voladora ausente de seres humanos” [6].

Otro término para agregar a la lista es el que precisó la agencia encargada de desarrollar nuevas tecnologías para uso militar del Departamento de Defensa de Estados Unidos conocida como Defense Advanced Research Projects Agency (DARPA). En el otoño del año 1996 DARPA definió la expresión Micro Air Vehicle (MAV) como un sistema de vuelo autónomo que cabía en una mano, con una dimensión máxima de 6 pulgadas (15.24 cms), cuyo propósito era apoyar soldados en tareas de reconocimiento, vigilancia y en la exploración de lo que había detrás de una colina o alrededor de un edificio [17]. Con el paso del tiempo se construyeron MAVs que cumplían con esta descripción, pero a medida que aumentaron las aplicaciones militares se incrementaron los tamaños de estas naves, y no se promulgó una nueva expresión que reflejara el cambio en su dimensión, ni se modificó la definición que ya existía, lo que se presta para confusión, porque hoy se puede llamar MAV a un vehículo de 13 o más pulgadas que no cabe en una mano.

A pesar de que ahora no existe un consenso en la clasificación, para el propósito de este trabajo de grado podemos usar el término micro-UAV para referirnos a UAVs que pesan menos de 1 kg (2 libras), tienen cortos rangos de operación local y vuelan bajo, desde cerca de la superficie hasta los 152.4 m (500 pies) [45]. También podemos usar la expresión micro-UAS para referirnos a UAS que tienen un Maximum Take-Off Weight (MTOW) en el rango

comprendido entre 0 y 1 kg [16]. MTOW es el peso máximo de despegue de una nave teniendo en cuenta la aeronave, la carga útil y el combustible.

“No fue sino hasta que aparecieron las computadoras digitales livianas y pequeñas, la tecnología de navegación por inercia y finalmente la red satelital Global Positioning System (GPS), que la operación autónoma de aeronaves no tripuladas obtuvo una autonomía de vuelo a la par con la de un vehículo pilotado por un ser humano” [6].

La Ilustración 6 muestra los elementos que componen un UAS [6].

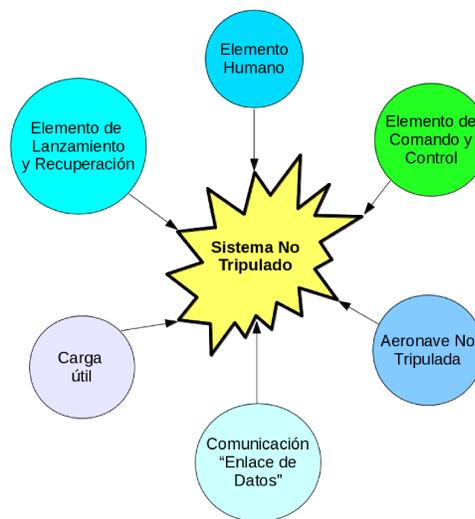


Ilustración 6.Elementos de un UAS [6].

2.5 Retos éticos y legales de los drones

En este momento, los drones son las armas preferidas para la guerra [5] porque:

1. Reducen los costos de entrenamiento y operación ya que es más barato entrenar pilotos de drones que pilotos de naves tripuladas, así como también es más económico operar una nave no tripulada que una tripulada. El ejército de Estados Unidos entrena ahora más pilotos de drones que de naves tripuladas.
2. Disminuyen los riesgos políticos porque reducen el número de bajas y capturas, por parte del enemigo.

3. Incrementan las capacidades militares de inteligencia, vigilancia, reconocimiento y apoyo aéreo, tanto a fuerzas desplegadas en tierra, como a misiones aéreas tripuladas.

4. Facilitan los asesinatos selectivos.

“Un **asesinato selectivo** es el uso intencional, premeditado y deliberado de la fuerza letal por parte de los Estados o sus agentes actuando bajo una apariencia legal, o por un grupo armado organizado en un conflicto armado, contra un individuo específico que no está bajo la custodia física del perpetrador. En los últimos años algunos Estados han adoptado políticas, bien sea abierta o implícitamente, de realizar asesinatos selectivos incluso en los territorios de otros Estados” [1]. Uno de los métodos de selección frecuentemente usado es el de localizar a la víctima por medio de su teléfono celular para dispararle desde el aire un misil con relativa precisión [40]. Cuando ocurren accidentes que resultan en daños a una propiedad superiores a 20.000 dólares o bajas humanas inocentes, los operadores del dron pueden reportar los mismos al interior de su organización [5]. Los asesinatos selectivos realizados por un Estado en el territorio de otro están prohibidos [1] a la luz del artículo 2(4) de la Carta de las Naciones Unidas que reza: “los Miembros de la Organización, en sus relaciones internacionales, se abstendrán de recurrir a la amenaza o al uso de la fuerza contra la integridad territorial o la independencia política de cualquier Estado, o en cualquier otra forma incompatible con los Propósitos de las Naciones Unidas”. Cuando los asesinatos selectivos violan el Derecho Internacional Humanitario entonces estos se podrían considerar crímenes de guerra [1]. Cuando los asesinatos selectivos se efectúan fuera de un conflicto armado entonces estos serían realmente ejecuciones extrajudiciales, extra legales, arbitrarias o sumarias [1].

Los robots en general afectan la privacidad de tres formas [13]:

1. Gracias a sus sensores y transmisores facilitan la vigilancia directa aumentando la visión y audición humana.

2. Facilitan el acceso a espacios históricamente protegidos como la propiedad privada o el territorio de una nación.

3. Tienen un significado social que los hacen llamar la atención de los seres humanos. Por ejemplo, los soldados tratan a sus drones esparcidos de bombas como si fueran compañeros y llegan incluso a arriesgar sus vidas para rescatar un dron atacado.

Posiblemente debido a que los seres humanos han desarrollado sus cerebros para ser expertos en la interacción social, las personas tienden a tratar a computadoras, tabletas, teléfonos celulares y robots como si fueran humanos. “A pesar de que nuestros cerebros han cambiado muy poco en miles de años, nosotros tenemos que tratar con la tecnología moderna. Como resultado, si una tecnología se comporta en una manera socialmente competente, nosotros evocamos nuestra evolucionada maquinaria social para interactuar con ella” [10]. Algunos creen que si la tecnología moderna presenta un comportamiento que despliega ricas habilidades sociales será aceptada por la mayoría de la gente [10]. Por esta razón existe una clara tendencia de diseño que busca construir robots humanoides y robots sociables. “En resumen, un robot sociable es inteligente socialmente en una forma semejante a la humana e interactuar con él es como interactuar con otra persona” [10].

Para Octubre de 2015 la Federal Aviation Administration (FAA) debe integrar completamente los drones gubernamentales, comerciales y de recreo al espacio aéreo de Estados Unidos como lo ordena la ley “FAA Modernization and Reform Act of 2012” (Public Law 112-95; 49 U.S.C. 40101 note). Debido a que los drones tienen la capacidad de transmitir información audiovisual de una persona sin su consentimiento y esta información se puede grabar y almacenar menoscabando el derecho a la intimidad, deberán aparecer nuevas leyes que regulen la operación de los drones de acuerdo a los principios de privacidad y exijan órdenes judiciales para poder espiar.

El uso de los drones presenta dos mitos que resumen claramente dos puntos de vista antagónicos que no ayudan a comprender los hechos que rodean la cotidiana realidad de los pilotos y operadores de drones. El mito heroico de los drones sostiene que esta tecnología incrementa las virtudes de los pilotos y operadores para hacer una guerra ética [5]. El mito antiheroico de los drones afirma que esta tecnología convierte a guerreros valientes y virtuosos en asesinos sin ética, o incluso en cobardes [5].

Generalmente la tripulación de un dron letal está conformada por el piloto, el operador de sensores y el coordinador de inteligencia de la misión [5]. Existen indicios de que la subjetividad de los pilotos y operadores está asociada a tres temas: la localización física, la seguridad personal y el prestigio social [5]. La localización física se refiere al efecto que produce la ausencia del cuerpo del piloto en la nave, que puede llevar a pensar que no es un piloto de verdad. La seguridad personal tiene que ver con la falta de exposición del cuerpo del piloto y los operadores a un ataque físico del enemigo, aunque sus conciencias ciertamente reciben un fuerte trauma psicológico al exponerse a escenas de guerra que generan un nuevo tipo de estrés denominado estrés virtual. Cuando algunos de ellos desarrollan trastornos por estrés postraumático son despedidos dando fin a su carrera militar. Un operador de drones debe repartir su tiempo entre operaciones remotas para matar insurgentes durante la noche y una típica vida doméstica durante el día. El prestigio social atañe a la falta de reconocimiento que percibe un piloto de drones cuando se compara con un piloto tradicional. Esta situación llevó a que la Fuerza Aérea de Estado Unidos le otorgara una insignia con alas de piloto a los pilotos de drones, así como lo hace con los pilotos tradicionales.

Las próximas generaciones de drones letales serán cada vez más autónomos y eliminarán la intervención humana en lo relacionado con su vuelo y la selección, localización y destrucción de sus objetivos [40]. Este nuevo tipo de drones pertenece a lo que se ha denominado sistema de arma autónoma. “Un **sistema de arma autónoma** es cualquier sistema que es capaz de elegir un blanco e iniciar el uso de la fuerza potencialmente letal sin supervisión e intervención humana directa en la toma de decisión letal” [4]. El especialista en robótica Ronald C. Arkin sugiere que este tipo de sistemas van a humanizar el campo de batalla porque pueden ser más éticos que los soldados humanos en el momento de usar la fuerza letal [3]. Por el contrario Jürgen Altmann, Noel Sharkey, Rob Sparrow y Peter Asaro creen que estos sistemas sólo pueden deshumanizar el campo de batalla y proponen su prohibición mundial, porque no se le debe permitir a las máquinas tomar la decisión de matar a las personas [4][41].

2.6 Parrot AR.Drone 1.0

La aeronave Parrot AR.Drone 1.0 se puede considerar un micro-UAS o micro-UAV de tipo VTOL. Por ser impulsado por cuatro rotores se denomina cuadricóptero (del inglés quadcopter) o cuadrirotor (del inglés quadrotor). En la Ilustración 7 se exhibe un Parrot AR.Drone 1.0 con su casco protector para vuelos en espacios interiores.



Ilustración 7. Parrot AR.Drone 1.0 con su casco protector para vuelos en espacios interiores.

El estudio del funcionamiento de este tipo de aeronaves ha permitido el desarrollo de varios modelos que han mejorado su estabilización y autonomía. Se conocen tres tipos de modelos posibles: los modelos de caja blanca (que nacen de la física del aparato), los modelos de caja gris (fruto de la física del aparato y de la optimización del modelo a la realidad) y los modelos de caja negra (que surgen de una total optimización del modelo a la realidad) [42]. Dentro de los modelos de caja blanca no lineales encontramos los de: Bouabdallah (2007) [9], Martínez (2007) [28] y Bresciani (2008) [11]. Dentro de los modelos de caja gris lineales está el de Pounds et al. (2002) [36]. Y dentro de los modelos de caja negra, que usan redes neuronales, está el de Sonntag (2011) [42].

En el campo de la educación y la investigación robótica el Parrot AR.Drone 1.0 ha sido usado en la navegación autónoma, el control de la posición, la suspensión en el aire sobre un objeto, la navegación apoyada en la visión, la localización y la vigilancia autónoma [25]. Uno de los últimos aportes es el de hacer funcionar el cuadrirotor en el medio ambiente de programación gráfica LabVIEW™ para hacer más fácil la entrada de los principiantes a la robótica aérea [29]. Un avance interesante es el obtenido en el vuelo autónomo en espacios interiores tales como: corredores, escaleras, cuartos y esquinas que se logró usando indicios de perspectiva y sonares de corto alcance montados al frente y a los lados de la nave. Para solucionar el problema, en lugar de construir modelos tridimensionales, se usó la información bidimensional entregada por las cámaras incorporadas. Utilizando dos algoritmos de OpenCV: el detector de bordes de Canny y la transformada probabilística de Hough, para encontrar las líneas largas, se consiguió detectar el punto de fuga de la imagen y hacer volar la aeronave en esa dirección [8]. En espacios exteriores sobresale la navegación autónoma realizada en bosques densos, usando solamente la cámara frontal del AR.Drone y un algoritmo que aprende a predecir el comportamiento que tendría un piloto humano en la misma situación de vuelo. Este sistema evita choques con árboles, ramas y arbustos a velocidades de hasta 1.5 m/s [39]. La navegación autónoma basada en la cámara frontal evolucionó hasta llegar a ambientes no estructurados y desconocidos, tanto en espacios interiores como en exteriores, gracias a una implementación que utilizó exclusivamente los sensores incorporados en el dron para estimar una escala absoluta en un mapa visual con el apoyo de tres elementos que funcionan armónicamente: un sistema SLAM (Simultaneous Localization And Mapping), de tipo visual, que aporta la localización y el mapa del entorno, un Filtro de Kalman Extendido que fusiona los datos de los sensores y predice el estado del cuadricóptero y un controlador PID (Proportional Integral Derivative) que genera los comandos para mover el cuadrirotor [18][19]. Otro salto importante en el dominio de la colaboración y coordinación entre vehículos terrestres no tripulados y vehículos aéreos no tripulados se dio con el desarrollo de un algoritmo que soluciona el problema de la protección a un convoy, asignando y despachando AR.Drones hacia amenazas y asegurándose de proteger a la caravana en todo momento [43].

En el campo de la seguridad informática el Parrot AR.Drone 1.0 ha sido usado para crear una botnet desde el cielo, agregando a su carga útil un pequeño Single Board Computer (SBC)

corriendo Debian Lenny Linux® para realizar ataques sigilosos a redes Wi-Fi® y separar el botmaster de sus bots. La aeronave puede navegar autónomamente o mediante una conexión celular 3G transformándose en un botmaster que envía y recibe los datos de comando y control por medio de las redes comprometidas [38].

Todos estos resultados han sido posibles gracias al bajo costo de este dron. Cuando se compara con cuadrirotos similares es uno de los más económicos (Tabla 1).

Nombre	Cámara	API	Open Source Software	Open Source Hardware	Costo
Parrot AR.Drone 1.0	Sí	Sí	No	No	300 dólares
DJI Phantom Quadcopter	No	No	No	No	679 dólares
3DR RTF Quad	Sí	Sí	Sí	Sí	935 dólares

Tabla 1. Comparación de algunos cuadricópteros similares.

Sin embargo una de las desventajas que tiene el Parrot AR.Drone 1.0 es que ni el software que corre a bordo, ni su hardware es fácilmente modificable porque no fue diseñado bajo modelos Open Source Software u Open Source Hardware. Esta situación restringe su uso a la recepción de comandos usando el protocolo y los canales de comunicación dispuestos por el fabricante usando una red Wi-Fi® [35].

2.7 Movimientos básicos del Parrot AR.Drone 1.0

Un Parrot AR.Drone 1.0 vuela porque sus cuatro rotores crean una fuerza de empuje que es superior a su peso (Ilustración 8). Una pareja de rotores gira en el sentido de las manecillas del reloj, mientras la otra gira en sentido contrario a las manecillas del reloj. El dron puede flotar en el aire porque la sumatoria de los cuatro torques creados por sus rotores suman cero y la fuerza de empuje compensa su peso.

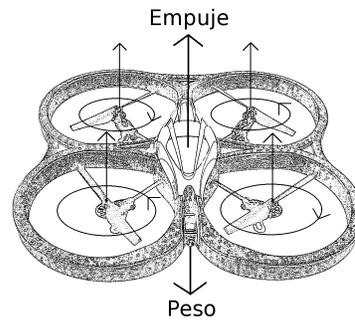


Ilustración 8.Fuerza de empuje.

Todos los movimientos básicos del cuadricóptero se pueden generar ajustando la velocidad de sus rotores. Por ejemplo, para ascender los cuatro rotores giran a una misma alta velocidad (Ilustración 9).

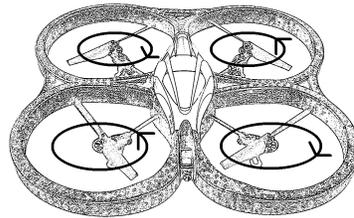


Ilustración 9.Ascenso.

Para descender los cuatro rotores giran a una misma baja velocidad (Ilustración 10).

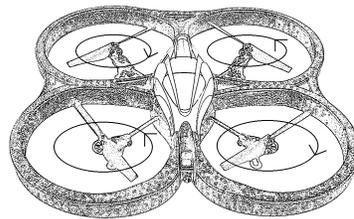


Ilustración 10.Descenso.

El movimiento horizontal se genera desnivelando una de las dos parejas de rotores. Una pareja mantiene igualdad de velocidad en sus rotores, mientras la otra desigualdad (Ilustración 11, Ilustración 12, Ilustración 13, Ilustración 14). La desigualdad se crea aumentando la velocidad de un rotor y disminuyendo la del otro.

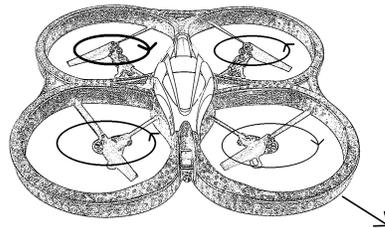


Ilustración 11.Movimiento 1.

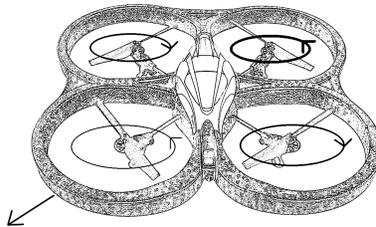


Ilustración 12.Movimiento 2.

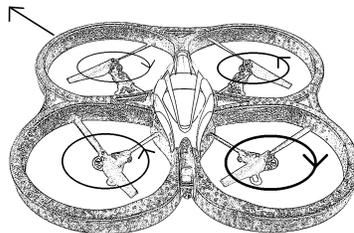


Ilustración 13.Movimiento 3.

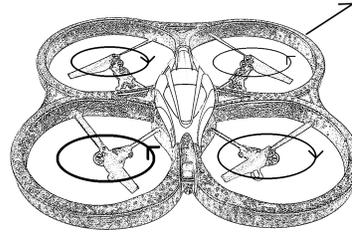


Ilustración 14.Movimiento 4.

Las dos rotaciones básicas se logran haciendo que una de las parejas mantenga una misma alta velocidad en sus dos rotadores, mientras la otra mantiene una misma baja velocidad en sus dos rotadores (Ilustración 15, Ilustración 16). Dependiendo de la forma en que se arreglan las parejas se produce una rotación en el sentido de las manecillas del reloj o en sentido contrario a las manecillas del reloj.

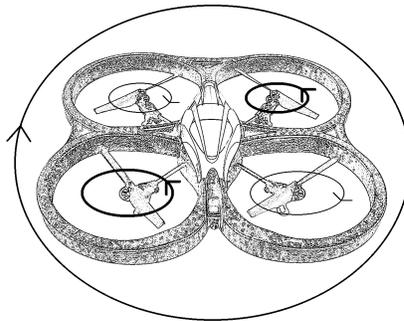


Ilustración 15.Rotación 1.

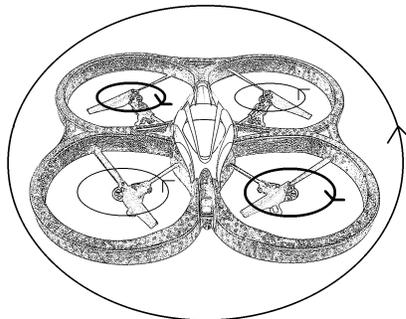


Ilustración 16.Rotación 2.

2.8 Comando de control de movimiento del Parrot AR.Drone 1.0

Para poder mover el cuadricóptero es necesario enviar un comando de control de movimiento desde cualquiera de los dispositivos compatibles.

Un comando de control de movimiento u se puede definir matemáticamente de la siguiente forma:

$$u = (\bar{\Phi}, \bar{\Theta}, \bar{z}, \bar{\Psi}) \in [-1, 1]^4$$

donde Φ es el ángulo de alabeo (Ilustración 17), Θ es el ángulo de cabeceo (Ilustración 18), \dot{z} es la velocidad vertical y $\dot{\Psi}$ es la velocidad angular de guiñada (Ilustración 19).

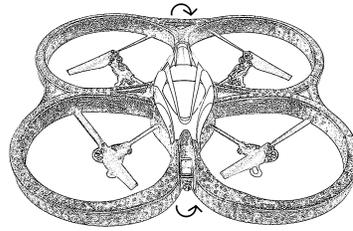


Ilustración 17.Alabeo.

El ángulo de alabeo Φ corresponde al ángulo de inclinación izquierda-derecha del dron y cuando es 0 significa plano horizontal, cuando es negativo indica inclinación hacia la izquierda, cuando es positivo indica inclinación hacia la derecha [35].

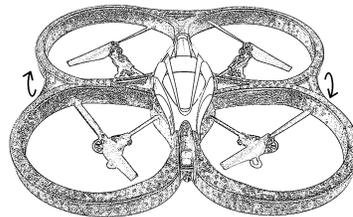


Ilustración 18.Cabeceo.

El ángulo de cabeceo Θ corresponde al ángulo de inclinación adelante-atrás del dron y cuando es 0 significa plano horizontal, cuando es negativo indica inclinación hacia adelante, cuando es positivo indica inclinación hacia atrás [35].

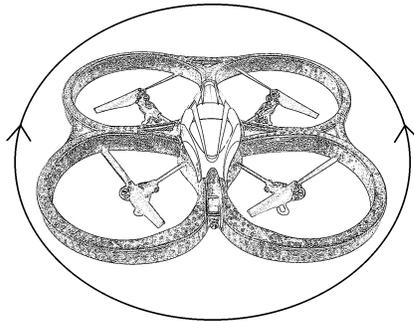


Ilustración 19.Guiñada.

El ángulo de guiñada Ψ corresponde al ángulo de rotación en el sentido de las manecillas del reloj o en sentido contrario a las manecillas del reloj. $\dot{\Psi}$ es el símbolo de la velocidad angular de guiñada.

2.9 Cámaras del Parrot AR.Drone 1.0

La cámara que mira al frente cubre un campo de visión de $73.5^\circ \times 58.5^\circ$ [19], tiene una resolución real de 320×240 píxeles, y una teórica de 640×480 píxeles [35]. Las imágenes se transmiten a una resolución de 320×240 debido a que con las primeras versiones del dron quedó demostrado que cuando se transmite el flujo de vídeo a una resolución de 640×480 la codificación del vídeo coloca mucha carga sobre el procesador afectando el vuelo del cuadrirotor [18]. El vídeo se transmite a un portátil a una tasa de 18 fps (fotogramas por segundo) usando compresión con pérdida [19].

La cámara que mira hacia abajo cubre un campo de visión de $47.5^\circ \times 36.5^\circ$ [19], tiene una resolución de 176×144 píxeles a una tasa real de 18 fps y una teórica de 60 fps [35]. El vídeo se transmite a un portátil a esa resolución y tasa real [18].

2.10 Giroscopios, acelerómetro y altímetro del Parrot AR.Drone 1.0

El Parrot AR.Drone 1.0 mide el cabeceo y el alabeo gracias a la fusión de los datos de dos de sus sensores: un acelerómetro de tres ejes y un giroscopio de dos ejes IDG-400 [25]. La desviación en la medición del cabeceo y el alabeo es baja y está entre 0.2° [25] y 0.5° [18].

El dron mide la guiñada por medio de un giroscopio de alta precisión XB-3500CV [25]. La deriva en la medición de la guiñada es alta y está entre 12° por minuto [25] y 60° por minuto [18], variando de un dron a otro.

El cuadrirotor mide la altura por medio de un transceptor-receptor ultrasónico que usa la técnica del sonar para cumplir la función de altímetro. El rango máximo de medición es de 6 metros de altura y la frecuencia de la medición es de 25 Hz. El error de este altímetro es superior al 20% y parece depender del tipo de superficie que se encuentra debajo del dron [25].

2.11 APIs para el Parrot AR.Drone 1.0

Parrot ofrece un Software Development Kit (SDK) escrito en ANSI C que permite desarrollar aplicaciones para Windows, iOS, Linux y Android [35].

Además algunos usuarios del cuadrirotor han creado APIs en otros lenguajes tales como C# (ARDrone-Control-.NET), Java (javadrone), Python (python-ardrone), JavaScript (node-ardrone) y Processing (ARDroneForP5, <http://kougaku-navi.net/ARDroneForP5/index.html>).

En Japón, Shigeo Yoshida del Departamento de Ingeniería de la Universidad de Tokio creó la librería para Processing ARDroneForP5 [47] y la liberó bajo una Licencia Apache.

2.12 DFRobotShop Rover V2 (Xbee® Kit)

El robot móvil DFRobotShop Rover V2 (Xbee® Kit) utiliza una caja de cambios de doble motor, una oruga y un juego de ruedas marca Tamiya, incorpora en su tarjeta de circuito impreso un Arduino Uno USB Rev 3, un controlador de motor L293B (conectado a los pines 5 al 8), un regulador de voltaje y un área para prototipos que permite construir y soldar circuitos. El robot se puede alimentar con un voltaje comprendido en el intervalo [3.7,9]¹. Tiene 2 módulos Xbee® (uno para montar en el robot y otro para conectar a la computadora con un breakout board USB a Xbee®). El robot se compra desarmado y se debe ensamblar y soldar. El fabricante permite ampliar el área de superficie del rover agregando una placa plástica de expansión.

2.13 API para el DFRobotShop Rover V2 (Xbee® Kit)

El rover se puede programar en el lenguaje Arduino C, un subconjunto del lenguaje ANSI C usando el IDE de Arduino (<http://arduino.cc/>). RobotShop Distribution Inc. ofrece el código de dos sketches en su página web (<http://www.robotshop.com/>) buscando por el producto DFRobotShop Rover V2. Uno de los ejemplos sirve para probar la velocidad máxima hacia el frente y el otro para mover el rover adelante, atrás, a la izquierda y a la derecha usando las teclas w,a,s,d de una computadora. Este último fue el que se usó en este trabajo de grado para controlar el rover. Ambos sketches aparecen escritos en el manual del usuario y también se pueden descargar en un archivo comprimido zip.

2.14 Teleoperación

Teleoperación significa “realizar algún trabajo o acción desde alguna distancia. Este término se asocia comúnmente a la robótica y a los robots móviles e indica el manejo de una de estas máquinas desde un lugar alejado del sitio que ocupa la misma” [7].

Por otra parte existen tres arquitecturas de control para los sistemas telerobóticos: el control directo, el control compartido y el control supervisado. El control directo es aquel en donde el robot no tiene inteligencia o autonomía y el usuario lo controla manualmente sin ninguna ayuda automatizada. El control compartido es aquel en donde el robot cuenta con alguna inteligencia o autonomía y el usuario tiene alguna ayuda automatizada como puede ser una realimentación aumentada. El control supervisado es aquel en donde el robot tiene una gran inteligencia o autonomía y el usuario se dedica a supervisar el trabajo del robot [31].

2.15 Augmented Reality (AR)

Augmented Reality (AR) se define como: “una visión directa o indirecta, en tiempo real, de un medio ambiente físico del mundo real, que ha sido mejorada/aumentada al agregarle información virtual generada por computadora” [21].

La AR no está limitada al sentido de la vista. “La AR puede aplicarse potencialmente a todos los sentidos, aumentando el olfato, el tacto y el oído también. La AR puede usarse para aumentar o substituir los sentidos que faltan en los usuarios por reemplazo sensorial, como por ejemplo: aumentar la vista de los usuarios invidentes o de los usuarios con visión

disminuida por medio del uso de señales auditivas, o aumentar el oído a usuarios sordos por medio del uso de señales visuales” [21].

Paul Milgram y Fumio Kishino definieron el “continuo realidad-virtualidad de Milgram” [21], como se muestra en la Ilustración 20, para poder visualizar la AR, la Realidad Virtual y la Realidad Mixta.



Ilustración 20. Continuo realidad-virtualidad de Milgram [21].

Un gran aporte a la AR surgió al final de los años noventa cuando Hirokazu Kato presentó una poderosa librería de herramientas, útil para crear aplicaciones visuales de AR asistida por computadora denominada ARToolKit [30].

2.16 Marcadores visuales

“Los marcadores visuales son símbolos gráficos diseñados para ser fácilmente reconocidos por máquinas” [15].

“Por ejemplo, los marcadores ARToolKit (Ilustración 21) se caracterizan por un marco cuadrado negro y grueso que contiene patrones arbitrarios en la escala de grises. El sistema usa la detección de líneas rectas y el encaje del cuadrilátero para localizar los marcadores y si un marcador es hallado, su patrón es extraído y comparado con todos los patrones conocidos. Como consecuencia de esto la velocidad del sistema disminuye entre más patrones se usen y entre más marcadores estén contenidos en la imagen de entrada. Los patrones usados para los marcadores ARToolKit pueden ser personalizados arbitrariamente, sin embargo, investigación reciente sugiere aplicar técnicas de codificación de comunicación digital para mejorar el desempeño del sistema, a costa del uso de la personalización” [15].



Ilustración 21. Marcador ARToolkit.

Los marcadores ARToolkit, colocados sobre el piso y en ciertos puntos conocidos de un cuarto, han sido usados para calcular la localización de un Parrot AR.Drone 1.0 con respecto a un punto de origen predeterminado [22]. Un algoritmo de seguimiento visual mantiene flotando el dron sobre el marcador ARToolkit que aparece en el centro de la imagen de la cámara. Esto permite actualizar la posición del cuadrirotor y estimar su posición relativa, al marcador, y absoluta, al punto de origen.

2.17 APIs para la AR

ARToolkit es la librería original escrita por Hirokazu Kato en C y C++ y es muy importante porque fue la base para la creación de las posteriores librerías para la AR. Se creó como una librería de código abierto y se liberó bajo una Licencia Pública General de GNU, por el Laboratorio de Tecnología de Interfaz Humana (HITLab) de la Universidad de Washington. Ahora, la ARToolkit es conservada por el HITLab y el HITLabNZ en la Universidad de Canterbury, Nueva Zelanda [30].

NyARToolkit (<http://nyatla.jp/nyartoolkit/wp/>) es una ARToolkit escrita en Java. Está basada en la ARToolkit 2.72.1 y tiene derivaciones para C++, C#, Silverlight, Action Script 3 (Flash10), Android, Unity3D y Processing.

En Japón, Ryo Iizuka creó la librería para Processing NyAR4psg y la liberó bajo una Licencia Pública General de GNU.

2.18 Controlador PID

“Un **controlador Proporcional Integral Derivativo (PID)** es un mecanismo genérico de control por realimentación ampliamente utilizado en sistemas de control industrial” [18]. Un regulador PID se usó en este trabajo de grado para controlar el dron. “Está basado en tres mecanismos separados de control, la señal de control es una suma ponderada de los siguientes tres términos:

- la parte **proporcional** depende del error presente $e(t)$,
- la parte **integral** depende del error pasado acumulado $\int_0^t e(\tau) d\tau$,
- la parte **derivativa** depende del error futuro predicho, basado en la derivada del error con respecto al tiempo $\dot{e}(t)$.

Si la integral y la derivada del error no pueden ser medidas directamente, ellas pueden ser aproximadas por medio de técnicas de derivación e integración numéricas:

$$\int_0^t e(\tau) d\tau \approx \sum_{\tau=1}^t e(\tau)$$

y

$$\dot{e}(t) \approx \frac{1}{\delta_t} (e(t) - e(t - \delta_t)) .$$

El controlador PID calcula los valores de entrada del sistema de acuerdo a :

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t)$$

donde K_p , K_i y K_d son parámetros ajustables que típicamente se determinan experimentalmente por medio del método de ensayo y error, sin embargo existen métodos heurísticos y guías para ayudar a desarrollar este proceso” [18].

2.19 Pseudocódigo para implementar un controlador PID

El pseudocódigo para implementar el algoritmo de un controlador PID es el siguiente:

lea la medición;

$e_{nuevo} = referencia - medición;$

$e_{punto} = e_{nuevo} - e_{viejo};$

$E = E + e_{nuevo};$

$u = K_p \cdot e_{nuevo} + K_i \cdot E + K_d \cdot e_{punto};$

$e_{viejo} = e_{nuevo};$

donde K_p , K_i y K_d incluyen la muestra de tiempo δ_t y E es la sumatoria de los errores.

2.20 Filtro Paso Bajo de primer orden

Un **Filtro Paso Bajo de primer orden**, también conocido como **filtro de media móvil ponderada exponencialmente** es aquel “que pasa la señal de baja frecuencia pero bloquea la señal de alta frecuencia. Es ampliamente usado para filtrar el ruido porque en muchos casos la señal a medir está en las bandas de baja frecuencia y el ruido está en las bandas de alta frecuencia” [24]. También se usa cuando se necesita obtener la tendencia de los datos a largo plazo [24]. Un Filtro Paso Bajo de primer orden se usó en este trabajo de grado para reducir el ruido del altímetro del dron.

Un filtro paso bajo de primer orden se expresa por medio de la siguiente expresión recursiva [24]:

$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1 - \alpha)x_k$$

donde α es una constante en el intervalo: $(0,1)^1$.

2.21 Pseudocódigo para implementar un Filtro Paso Bajo de primer orden

El pseudocódigo para implementar el algoritmo de un Filtro Paso Bajo de primer orden es el siguiente:

lea la entrada ;

estimado = $\alpha \cdot \text{medicionPrevia} + (1.0 - \alpha) \cdot \text{entrada} ;$

medicionPrevia = estimado ;

donde $\alpha \in (0,1)^1$.

III – DESARROLLO DEL TRABAJO

El proceso de construcción del prototipo se realizó utilizando la metodología del **modelo unificado de diseño** tratado en la sección 2.4 del capítulo I.

1. Fase del ciclo de diseño

1.1 Selección de sensores

Del control remoto Wii™ se escogieron los botones: up, down, left, right, a, b, minus, plus, home, one y two para pilotar el dron manualmente y ponerlo a flotar sobre el rover.

Del Parrot AR.Drone 1.0 se eligieron: la cámara que mira hacia abajo para que observe al rover, el altímetro para que conserve la altura constante sobre el rover, el acelerómetro de tres ejes y el giroscopio de dos ejes IDG-400 para vigilar el cabeceo y el alabeo de la nave sobre el rover.

1.2 Selección de librerías

Para conectar el control remoto Wii™ vía Bluetooth® al portátil se tomó la única opción disponible: wrj4p5 (<http://sourceforge.jp/projects/wrj4p5/>).

Para conectar el Parrot AR.Drone 1.0 vía Wi-Fi® al portátil se aprovechó la única opción disponible: ARDroneForP5 (<http://kougaku-navi.net/ARDroneForP5/index.html>).

Para usar la realidad aumentada se empleó la única opción disponible: NyAR4psg (<http://nyatla.jp/nyartoolkit/wp/>).

Para crear la caja negra del dron y grabar las imágenes capturadas sobre el rover se utilizó la librería de Andrés Colubri: GSVideo (<http://gsvideo.sourceforge.net/>), liberada bajo Licencia Pública General Reducida de GNU.

1.3 Selección de estrategia

Para que el dron pueda seguir al rover, usando solamente el flujo de vídeo de su cámara que mira hacia abajo, es necesario detectar los dos movimientos independientes presentes en una misma imagen: el del dron y el del rover.

Como no se cuenta con un marco de referencia absoluto que informe la posición del dron y del rover, se puede manipular la imagen de la cámara para determinar la posición relativa de uno con respecto al otro. Para ello se coloca un marcador visual ARToolKit sobre el rover que permite conocer su posición dentro de la imagen.

Para que el dron siga al rover es necesario generar los movimientos en el dron que permitan alinear el centro de la imagen de la cámara del dron con el centro del marcador visual ARToolKit del rover.

En lo referente a la distribución de la visión aumentada se determinó mostrar una esfera amarilla sobre el marcador visual ARToolKit del rover.

1.4 Selección del algoritmo de detección, seguimiento y realidad aumentada

Luego de buscar los algoritmos de seguimiento visual que han sido empleados con los cuadrirotos Parrot AR.Drone 1.0 y los marcadores visuales ARToolKit se usó el único encontrado: mantener flotando el dron sobre el marcador ARToolKit que aparece en el centro de la imagen de su cámara vertical y actualizar su posición persiguiendo el marcador en las imágenes subsiguientes [22].

1.5 Definición del modelo para controlar el dron

El modelo que se adoptó para controlar el dron es:

$$\ddot{e} = k \cdot u - g$$

donde \ddot{e} es la aceleración del dron, k es un coeficiente desconocido, u es el comando de control de movimiento del cuadrirotor descrito en la sección 2.8 del capítulo II y g es la gravedad.

1.6 Proyección de los estados del prototipo

Para el prototipo se proyectaron sólo dos estados como se muestra en la Ilustración 22. El **control manual** para que un humano teleopere el dron con el control remoto Wii™ al inicio, al final y cuando el algoritmo de seguimiento visual pierda de vista el marcador. El **seguimiento automático** para que el portátil teleopere el dron sin asistencia humana.



Ilustración 22.Diagrama de estados del prototipo

1.7 Proyección de la clase FiltroPasoBajo

La clase FiltroPasoBajo (Ilustración 23) se planificó en este trabajo de grado para reducir el ruido del altímetro del dron.

FiltroPasoBajo
+ alfa : float
+ medicionPrevia : float
+ primeraEjecucion : bool
+ FiltroPasoBajo(a : float)
+ estimar(entrada : float) : float
+ reiniciar()

Ilustración 23.Diagrama de la clase FiltroPasoBajo

1.8 Proyección de la clase ControladorPID

La clase ControladorPID (Ilustración 24) se concibió en este trabajo de grado para que el portátil regulara automáticamente el cabeceo, el alabeo y la velocidad vertical del dron.

ControladorPID
+ Kp : float
+ Ki : float
+ Kd : float
+ integralDelError : float
+ errorViejo : float
+ MAXIMO_U : float
+ primeraEjecucion : bool
+ ControladorPID(p : float, i : float, d : float, maxu : float)
+ estimar(medicion : float, referencia : float) : float
+ reiniciar()

Ilustración 24. Diagrama de la clase ControladorPID

1.9 Pseudocódigo del algoritmo de detección, seguimiento y realidad aumentada

inicie grabación de vídeo en el archivo cajaNegra.ogg ;

inicie conexión Bluetooth® con el Wii™ ;

inicie configuración para la realidad aumentada ;

inicie conexión Wi-Fi® con el dron ;

haga por siempre :

si la carga de la batería < 18 por ciento haga vibrar el Wii™ ;

pinte el ícono de la batería en la pantalla ;

si se oprime el botón TWO del Wii™ active la cámara vertical ;

si se oprime el botón MINUS del Wii™ entre en estado de seguimiento automático y

pinte el ícono de seguimiento , un círculo blanco debajo del ícono de la batería ;

detecte el marcador visual ARToolkit ;

si pierde de vista el marcador visual ARToolKit entre en estado de control manual y haga estos 2 pasos :

1.reinicie los controladores PID de cabeceo y alabeo y el Filtro Paso Bajo de la altura si y sólo si estaba en estado de seguimiento , con la cámara vertical activada y volando ;

2.vuelva al inicio del ciclo haga por siempre ;

si no pierde de vista el marcador visual ARToolKit haga estos 5 pasos :

1. encuentre la coordenada X ,Y del centro del marcador , pinte una cabeza de flecha verde indicando la dirección hacia esa coordenada y calcule la distancia a ese punto X ,Y desde el centro de la imagen ;

2. estime la altura con el Filtro Paso Bajo aplicado a la señal del altímetro ;

3. si estaba en estado de seguimiento , con la cámara vertical activada y volando haga lo siguiente :

si la distancia al centro del marcador desde el centro de la imagen ≤ 15 pixeles haga flotar el dron ;

si la distancia al centro del marcador desde el centro de la imagen > 15 pixeles envíe el comando de control de movimiento (u) con el cabeceo , alabeo y velocidad vertical estimados por los respectivos controladores PID ;

4. si se oprime el botón HOME del Wii TM muestre una esfera amarilla sobre el marcador visual ARToolKit ;

5. agregue un nuevo fotograma con los pixeles de la pantalla al archivo cajaNegra.ogg ;

2. Fase de producción

2.1 Descripción del montaje

Los elementos que se emplearon para la construcción del prototipo fueron los siguientes: un portátil con Wi-Fi® y Bluetooth® corriendo el sistema operativo Slackware® Linux® 14 desarrollado por Patrick Volkerding, un control remoto Wii™, un Parrot AR.Drone 1.0, un DFRobotShop Rover V2 (Xbee® Kit), un marcador visual ARToolKit colocado sobre la placa plástica de expansión del rover, el IDE de Processing versión 2.1, el IDE de Arduino versión 1.0.5, un cuarto libre de obstáculos y con buena iluminación.

Todo el proyecto se desarrolló con la premeditada intención de usar hardware de bajo costo, disponible comercialmente, para facilitar la replicación del trabajo, y software de código abierto para promover el uso, cambio, mejoramiento y redistribución del código fuente para que la comunidad de la robótica aérea pueda realizar sin problemas sus investigaciones y desarrollos.

El portátil se conecta por Wi-Fi® con el dron, por Bluetooth® con el control remoto Wii™ y por Xbee® con el rover. El dron recibe los comandos de control de movimiento enviados por el portátil bien sea que el prototipo se encuentre en estado de seguimiento automático o en control manual y envía al portátil la información de sus cámaras y sensores de navegación. El control remoto Wii™ recibe los comandos enviados por el portátil y vibra en la mano del usuario para avisarle algo. También envía al portátil la información de sus sensores. El rover recibe comandos enviados por el usuario desde el portátil para moverse en una dirección o en otra y puede enviar al portátil información en caso de que se instalen y conecten sensores a su tarjeta de circuito impreso.

2.2 Instalación de las librerías de Processing

Se descargó el IDE de Processing versión 2.1 para Linux® de 32 bits de <http://processing.org/>. Dentro del directorio *sketchbook/libraries* se colocaron todas las librerías que se usaron. Para conocer la ubicación de esta carpeta se usó en el menú

File del IDE de Processing la opción *Preferences* que muestra la ubicación del sketchbook.

Para poder usar el control remoto Wii™ se descargaron los siguientes 5 archivos:

de <http://sourceforge.jp/projects/wrj4p5/> 2 archivos:

wrj4P5.jar

Loc.jar

de <https://github.com/micromu/WiiRemoteJ> 1 archivo:

WiiRemoteJ.jar (versión 1.6)

de <http://code.google.com/p/bluecove/downloads/list> 2 archivos:

bluecove-2.1.0.jar

bluecove-gpl-2.1.0.jar

Se creó esta ruta de carpetas *libraries/wrj4P5/library* dentro de la carpeta del *sketchbook*. Ubicado en el directorio *sketchbook* se impartieron los siguientes comandos del shell: **mkdir libraries**, **cd libraries**, **mkdir wrj4P5**, **cd wrj4P5**, **mkdir library**. La nueva ruta quedó así *sketchbook/libraries/wrj4P5/library*. Finalmente se copiaron los 5 archivos mencionados arriba en la ruta recién creada y listo.

Para poder usar el Parrot AR.Drone 1.0 se descargó el archivo *ARDroneForP5-master.zip* de <http://kougaku-navi.net/ARDroneForP5/index.html>, se descomprimió y se colocó la carpeta entera *ARDroneForP5-master* en la ruta *sketchbook/libraries*. Para que todo funcionara bien se descargaron y agregaron los archivos que se listan abajo a cada uno de los sketches de ejemplo de la librería usando en el menú *Sketch* del IDE de Processing la opción *Add File*.

De <http://www.xuggle.com/downloads> se descargaron y agregaron los siguientes 3 archivos:

xuggle-xuggler-noarch-5.4.jar

xuggle-xuggler-arch-i686-pc-linux-gnu.jar

xuggle-xuggler-5.4.jar .

De <http://www.slf4j.org/download.html> se descargaron y agregaron los siguientes 4 archivos:

slf4j-jdk14-1.7.5.jar

slf4j-jdk14-1.7.5-sources.jar

slf4j-api-1.7.5.jar

slf4j-api-1.7.5-sources.jar

De http://commons.apache.org/proper/commons-net/download_net.cgi se descargaron y agregaron los siguientes 3 archivos:

commons-net-examples-3.3.jar

commons-net-3.3.jar

commons-net-3.3-sources.jar

Para poder usar la realidad aumentada se descargó el archivo *nyar4psg-1.3.1.zip* de <http://nyatla.jp/nyartoolkit/wp/> , se descomprimió y se colocó la carpeta entera *nyar4psg-1.3.1* en la ruta *sketchbook/libraries* .

Para poder crear la caja negra del dron y grabar las imágenes capturadas sobre el rover se descargó el archivo *GSVideo-1.0.0-linux.zip* de <http://gsvideo.sourceforge.net/> , se descomprimió y se colocó la carpeta entera *GSVideo-1.0.0-linux* en la ruta *sketchbook/libraries* .

2.3 Prueba inicial de funcionamiento

Se probaron cada uno de los elementos que se emplearon para la construcción del prototipo de la siguiente forma.

En el caso del control remoto Wii™ se estableció la comunicación con el portátil vía Bluetooth®. Con el comando del shell **hciconfig hci0 up** se activó la interfaz Bluetooth® de la computadora. Se asignaron los botones del control remoto a algunos comandos básicos del dron como despegar, aterrizar, moverse adelante, atrás, a la derecha y a la izquierda.

En el caso del DFRobotShop Rover V2 (Xbee® Kit) se debió realizar primero la soldadura y ensambladura del mismo. Se preparó el área de conducción, se cargó el sketch del fabricante que permite usar las teclas w,a,s,d para mover el rover adelante, atrás, a la izquierda y a la derecha. Se usó el comando del shell **picocom --echo --imap crcrLf /dev/ttyACM0** para poder realizar la comunicación serial entre el módulo Xbee® del rover y el módulo Xbee® del portátil.

En el caso del Parrot AR.Drone 1.0 se estableció la comunicación con el portátil vía Wi-Fi® conectando la computadora a la red que ofrece el dron. Se preparó el área de vuelo y se pilotó la nave usando primero las teclas del portátil y luego los botones del control remoto Wii™. Se experimentó con la realidad aumentada colocando un marcador visual ARToolKit sobre el piso y usando el vídeo enviado por el dron para agregar una esfera amarilla a la escena.

2.4 Implementación del prototipo

Se colocó sobre la placa plástica de expansión del rover un marcador visual ARToolKit de 8cmX8cm como se aprecia en la Ilustración 25. Debido a que la cámara vertical del dron no pudo reconocer fácilmente un marcador tan pequeño se colocó uno más grande impreso en una hoja tamaño carta sobre un cartón adherido con cinta de enmascarar como se observa en la Ilustración 26.

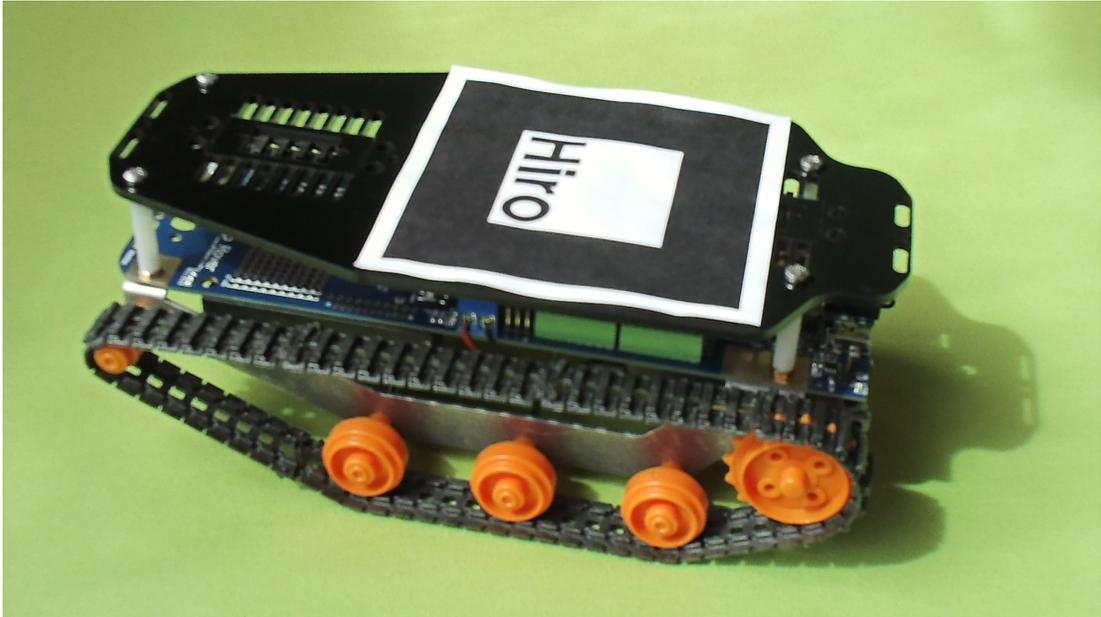


Ilustración 25.Rover con marcador visual ARToolKit de 8cmX8cm

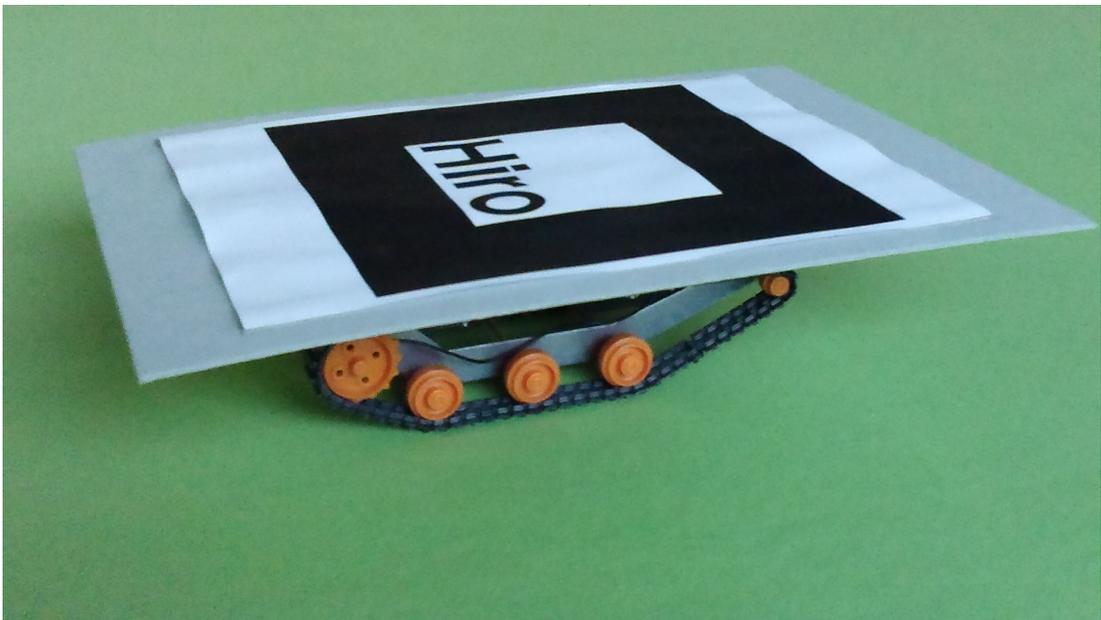


Ilustración 26.Rover con marcador visual ARToolKit tamaño carta

Lo primero que se creó fue la caja negra del dron para registrar en un vídeo las imágenes transmitidas por la cámara vertical y así poder analizar y depurar el desarrollo de la implementación.

Para el desarrollo del prototipo se consideraron dos casos concretos: el primero fue con el rover en estado de reposo y el segundo con el rover en movimiento. Para el primero el dron debía mantenerse flotando encima del rover y para el segundo el dron debía perseguir al rover. Se estableció el primer caso como punto de partida para resolver fácilmente el segundo.

Para resolver ambos casos se diseñaron las clases FiltroPasoBajo y ControladorPID descritas en la sección 1.7 y 1.8 del capítulo III respectivamente. Como con respecto al piso la superficie del vehículo terrestre no tripulado es más alta, cuando el dron volaba sobre el rover su transceptor-receptor ultrasónico generaba mucho ruido porque cuando medía la altura a ratos medía la altura con respecto al piso y a otros con respecto al rover. Se eliminó parte del ruido con la clase FiltroPasoBajo. Posteriormente se creó una instancia de la clase ControladorPID dedicada a regular la velocidad vertical y hacer que el dron mantuviera una altura de 1 metro relativamente constante sobre el rover. Después se crearon dos instancias más de la clase ControladorPID dedicadas a regular el cabeceo y el alabeo del dron para hacerlo volar alrededor y sobre el centro del marcador visual ARToolKit del rover. Usando el método de ensayo y error se ajustaron los parámetros K_p , K_i y K_d de cada una de las tres instancias de la clase ControladorPID para mantener al dron flotando encima del rover.

El segundo caso se resolvió asignando los siguientes valores a los parámetros ajustables de los controladores PID. Para el cabeceo $K_p=0.001$, $K_i=0.00001$ y $K_d=0.03$, para el alabeo: $K_p=0.001$, $K_i=0.00001$ y $K_d=0.03$. Para la velocidad vertical $K_p=0.001$, $K_i=0.000001$ y $K_d=0.001$. El ruido del altímetro del dron se manejó con un Filtro Paso Bajo de primer orden con un $\alpha=0.9$.

2.5 Sorpresas durante la implementación del prototipo

Durante el desarrollo del prototipo ocurrieron los siguientes eventos imprevistos:

- Fracturas del casco protector del dron. Debido a los frecuentes golpes contra las paredes del cuarto, el casco se fracturó varias veces y se reparó soldándolo con una pistola de silicona caliente. Para que no quedara exceso de silicona alrededor de la parte soldada se usó el dedo índice humedecido con agua para retirar el exceso de material, sin quemarse y dejando la superficie pegada lisa.

- Excepción de la librería *NyAR4psg*. La librería de vez en cuando genera una excepción que hace colapsar cualquier aplicación desarrollada con ella. Cuando se producía esta excepción dejaba al dron flotando en el aire y sin ninguna comunicación. Para hacerlo aterrizar tocaba pulsar el botón *Stop*, seguido del botón *Run* en el IDE de Processing.

- Estrelladas, caídas y movimientos inesperados del dron. Durante todo el tiempo de desarrollo se presentaron estos eventos, especialmente durante el ajuste de los parámetros K_d de los tres controladores PID diseñados para regular el cabeceo, el alabeo y la velocidad vertical.

2.6 Prueba del prototipo

El prototipo se probó realizando muchos vuelos de seguimiento automático en el cuarto destinado a ello. Medir exactamente la distancia de desviación del dron alrededor del rover no fue posible debido a que no se tenía un sistema absoluto de referencia que permitiera medir la distancia entre el dron y el rover. Sin embargo se diseñó una sencilla prueba para tratar de medir esta distancia, al menos aproximadamente.

Con un marcador borrable se pintó una cuadrícula sobre el piso del cuarto de vuelo conformada por cuadrados de 10cm de lado como se exhibe en en la Ilustración 27.



Ilustración 27. Cuadrícula pintada sobre el piso del cuarto de vuelo

Se realizaron 9 vuelos seguidos del dron siguiendo al rover. Se conservaron los 9 vídeos y se extrajeron los fotogramas de cada vídeo. Se analizaron, una tras otra, cada una de las fotos para medir el alejamiento de la cabeza de flecha verde al centro del marcador visual teniendo en cuenta la cuadrícula como se muestra en la Ilustración 28.



Ilustración 28.Fotograma del vídeo de la caja negra del dron

Se observó que la cabeza de flecha verde se mueve a una distancia aproximada de hasta 30cm alrededor del centro del marcador visual ARToolKit del rover. Para comparar el resultado se realizó el mismo experimento en un lugar más grande y con influencia de suaves corrientes de aire y se observaron distancias mayores.

2.7 Comunicación de resultados

Se comunicaron los resultados del prototipo en la memoria de trabajo de grado y en la página web.

2.8 ¿ Cómo se usa el prototipo ?

Primero se debe conectar el portátil a la red Wi-Fi® que ofrece el dron. Luego se debe activar la interfaz de Bluetooth® del portátil con el comando del shell **hciconfig hci0 up**.

En el IDE de Processing se debe abrir el sketch del prototipo y se debe oprimir el botón *Run* . Una vez se hace esto aparece una imagen como la que se muestra en la Ilustración 29 solicitando pulsar a la vez los botones ONE y TWO del control remoto Wii™ para establecer la conexión entre el portátil y el Wii™. Luego aparece en la pantalla la imagen que envía la cámara frontal del dron y en la esquina superior derecha aparece el ícono de carga de la batería.



Ilustración 29.Imagen inicial del prototipo

Para despegar el dron se pulsa en el Wii™ el botón PLUS. Luego se activa la cámara vertical con el botón TWO y se muestra en la pantalla la imagen que envía la misma. Usando los botones UP, DOWN, LEFT, RIGHT y B se pilota manualmente el dron hasta colocarlo sobre el rover. Una vez se encuentra encima se pulsa el botón MINUS para activar el seguimiento automático y aparece en la pantalla el ícono de seguimiento, que es un círculo blanco debajo del ícono de la batería, como se puede apreciar en la Ilustración 30.

El dron flotará sobre el rover y si este se mueve lo perseguirá gracias a los comandos de control de movimiento que le envía el portátil. Si se desea activar la función de realidad aumentada se pulsa el botón HOME en el Wii™ y aparece una esfera amarilla encima del marcador visual ARToolKit como se observa en la Ilustración 31. Si se desea aterrizar se pulsa el botón A cuidando de que no lo haga encima del rover.

En caso de que se pierda de vista el rover se puede pilotar manualmente el dron hasta colocarlo nuevamente sobre el rover. Si la carga de la batería del dron es menor al 18 por ciento el control remoto Wii™ vibra para que el piloto aterrice la nave antes de que se desplome por falta de energía.



Ilustración 30.Ícono de seguimiento debajo del ícono de carga de la batería



Ilustración 31.Esfera amarilla sobre el marcador visual ARToolKit

Los botones del control remoto Wii™ están asignados de la siguiente forma:

UP mueve el dron hacia adelante

DOWN mueve el dron hacia atrás

LEFT mueve el dron hacia la izquierda

RIGHT mueve el dron hacia la derecha

A aterriza el dron

B para el dron

MINUS activa el seguimiento automático

PLUS despega el dron

HOME activa la realidad aumentada

ONE activa la cámara horizontal

TWO activa la cámara vertical

IV - RESULTADOS Y REFLEXIÓN SOBRE LOS MISMOS

1. Cumplimiento del objetivo general

El objetivo general fue desarrollar un prototipo de software que permitiera teleoperar un cuadricóptero Parrot AR.Drone 1.0 en tareas de detección y seguimiento automático de un DFRobotShop Rover V2 previamente etiquetado con un marcador visual ARToolKit en un área de vuelo libre de obstáculos, ubicada en un espacio interior con buena iluminación.

El objetivo general planteado se realizó. Con el prototipo de software elaborado se logró mantener el dron a una altura aproximada de 1 metro siguiendo al rover a una distancia de hasta 30 centímetros alrededor del mismo.

Usando un Filtro Paso Bajo de primer orden para filtrar el ruido del altímetro del dron y unos controladores PID para generar los movimientos de cabeceo, alabeo y velocidad vertical se pudo alinear el centro de la imagen de la cámara del dron con el centro del marcador visual ARToolKit del rover. Se encontraron los parámetros ajustables de los controladores PID que permitieron alcanzar el seguimiento automático. Para el cabeceo $K_p=0.001$, $K_i=0.00001$ y $K_d=0.03$,para el alabeo: $K_p=0.001$, $K_i=0.00001$ y $K_d=0.03$. Para la velocidad vertical $K_p=0.001$, $K_i=0.000001$ y $K_d=0.001$. El ruido del altímetro del dron se manejó con un Filtro Paso Bajo de primer orden con un $\alpha=0.9$.

2. Cumplimiento de los objetivos específicos

En la Tabla 2 se muestran las dos fases planteadas, con sus objetivos específicos y sus estados de ejecución.

Fase	Objetivo específico	Estado	Observaciones
<i>Fase del ciclo de diseño</i>	Seleccionar el algoritmo que se empleó para que el robot aéreo detectara, siguiera y mostrara la información del objeto marcado usando realidad aumentada.	[Hecho]	El desarrollo de esta fase y su objetivo específico se describe en la toda la sección 1 del capítulo III
<i>Fase de producción</i>	Implementar el prototipo de software con el algoritmo elegido y considerando el uso de marcadores visuales ARToolKit para demostrar su funcionamiento.	[Hecho]	El desarrollo de esta fase y su objetivo específico se describe en la toda la sección 2 del capítulo III

Tabla 2. Estados de ejecución de las fases y sus objetivos específicos

3. Aporte a la solución del problema

La estrategia y el algoritmo desarrollado en este trabajo de grado se pueden ajustar a otros modelos de drones.

4. Impacto presente y futuro

Este trabajo de grado deja una huella cultural para que se sigan construyendo proyectos usando hardware comercial de bajo costo y software de código abierto que promueva el uso, cambio, mejoramiento y redistribución del código fuente entre los miembros de una comunidad interesada en compartir sus desarrollos sin restricciones.

5. Mejoras del prototipo

- Se deben investigar e implementar otras técnicas que permitan mejorar el mantenimiento de la altura del dron sobre el rover.
- Se puede diseñar e implementar un filtro de Kalman que reciba como entrada la coordenada X,Y del centro del marcador visual ARToolKit y genere como salida la estimación de la posición real dentro de la imagen que envía la cámara vertical.

V – CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

1. Conclusiones

- El objetivo general planteado se realizó porque con el prototipo de software elaborado se logró mantener el dron a una altura aproximada de 1 metro siguiendo al rover a una distancia de hasta 30 centímetros alrededor del mismo.
- La estrategia y el algoritmo desarrollado en este trabajo de grado se pueden ajustar a otros modelos de drones.

2. Recomendaciones

- Los estudiantes que deseen elaborar trabajos de grado con drones deben considerar usar software y hardware de código abierto.

3. Trabajos Futuros

- Diseñar e implementar otros prototipos usando OpenCV.
- Diseñar e implementar otros prototipos que no requieran marcadores visuales.

VI - REFERENCIAS

Este trabajo de grado utilizó la norma ACM porque es la que usa la Association for Computing Machinery (ACM) para citar la literatura de computación.

[1] ALSTON, P. Report of the special rapporteur on extrajudicial, summary or arbitrary executions: Addendum: Study on targeted killings. United Nations Human Rights Council, A/HRC/14/24/Add.6, May 2010.

[2] ALTROCK, R. C. The progress of solar cycle 24 at high latitudes. In *SOHO-23: Understanding a Peculiar Solar Minimum* (June 2010), S. R. Cranmer, J. T. Hoeksema, & J. L. Kohl, Ed., vol. 428 of *Astronomical Society of the Pacific Conference Series*, pp. 147-152.

[3] ARKIN, R. *Governing Lethal Behavior in Autonomous Robots*, 1st ed. Chapman & Hall/CRC, 2009.

[4] ASARO, P. On banning autonomous weapon systems: human rights, automation, and the dehumanization of lethal decision-making. *International Review of the Red Cross* 94, 886 (2012), 687-709.

[5] ASARO, P. The labor of surveillance and bureaucratized killing: new subjectivities of military drone operators. *Social Semiotics* 23, 2 (2013), 196-224.

[6] BARNHART, R., SHAPPEE, E., Y DOUGLAS M. MARSHALL, J. *Introduction to Unmanned Aircraft Systems*. Taylor and Francis, 2011.

[7] BASAÑEZ, L., Y SUÁREZ, R. Teleoperation. In *Springer Handbook of Automation*, S. Y. Nof, Ed. Springer Berlin Heidelberg, 2009, pp. 449-468.

[8] BILLS, C., CHEN, J., Y SAXENA, A. Autonomous mav flight in indoor environments using single image perspective cues. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (may 2011), pp. 5776 -5783.

- [9] BOUABDALLAH, S. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Vaud, Switzerland, February 2007.
- [10] BREAZEAL, C. *Designing Sociable Robots*, second ed. Intelligent Robotics and Autonomous Agents series. A Bradford Book. MIT Press, Cambridge, MA, August 2004.
- [11] BRESCIANI, T. *Modelling, identification and control of a quadrotor helicopter*, vol. 5823. Department of Automatic Control, Lund University, Lund, Scania, Sweden, 2008.
- [12] BRISTEAU, P. J., CALLOU, F., VISSIERE, D., Y PETIT, N. The navigation and control technology inside the ar.drone micro uav. In *18th IFAC World Congress* (Milano, Italy, 2011), pp. 1477-1484.
- [13] CALO, M. R. *Robot Ethics: The Ethical and Social Implications of Robotics*. Intelligent Robotics and Autonomous Agents series. A Bradford Book. MIT Press, Cambridge, MA, December 2011, ch. Robots and Privacy, pp. 187-201.
- [14] CHERTOFF, D. B., BYERS, R. W., Y LAVIOLA, JR., J. J. An exploration of menu techniques using a 3d game input device. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (New York, NY, USA, 2009), FDG '09, ACM, pp. 256-262.
- [15] COSTANZA, E., Y HUANG, J. Designable visual markers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, ACM, pp. 1879–1888.
- [16] DALAMAGKIDIS, K., VALAVANIS, K., Y PIEGL, L. Current status and future perspectives for unmanned aircraft system operations in the us. *Journal of Intelligent and Robotic Systems* 52 (2008), 313-329.
- [17] DARPA. *DARPA: 50 Years of Bridging the Gap*. Faircount LLC, 2008.

- [18] ENGEL, J. Autonomous camera-based navigation of a quadrocopter. Master's thesis, Technical University Munich, Germany, Dec. 2011.
- [19] ENGEL, J., STURM, J., Y CREMERS, D. Camera-based navigation of a low-cost quadrocopter. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)* (Oct. 2012).
- [20] FAA. Unmanned aircraft systems (uas) operational approval. Federal Aviation Administration, Notice 8900.227, July 2013.
- [21] FURHT, B. *Handbook of Augmented Reality*. Springer Science+Business Media. Springer, 2011.
- [22] JAYATILLEKE, L., Y ZHANG, N. Landmark-based localization for unmanned aerial vehicles. In *Systems Conference (SysCon), 2013 IEEE International* (2013), pp. 448–451.
- [23] KAJASTILA, R. A., Y LOKKI, T. A gesture-based and eyes-free control method for mobile devices. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems* (New York, NY, USA, 2009), CHI EA '09, ACM, pp. 3559-3564.
- [24] KIM, P. *Kalman Filter for Beginners: with MATLAB Examples*, 1st ed. CreateSpace, Lexington, KY, 2011.
- [25] KRAJNIK, T., VONASEK, V., FISER, D., Y FAIGL, J. Ar-drone as a platform for robotic research and education. In *Research and Education in Robotics - EUROBOT 2011*, D. Obdrzalek and A. Gottscheber, Eds., vol. 161 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2011, pp. 172-186.
- [26] LAVIOLA, JR., J. J., Y MARKS, R. L. An introduction to 3d spatial interaction with video game motion controllers. In *ACM SIGGRAPH 2010 Courses* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 2:1-2:78.

- [27] LEE, J. Hacking the nintendo wii remote. *Pervasive Computing, IEEE* 7, 3 (july-sept. 2008), 39-45.
- [28] MARTINEZ, V. Modelling of the flight dynamics of a quadrotor helicopter. Master's thesis, Cranfield University, Cranfield, Bedfordshire, England, United Kingdom, September 2007.
- [29] MOGENSEN, M. The ar drone labview toolkit: A software framework for the control of low-cost quadrotor aerial robots. Master's thesis, Tufts University, Massachusetts, MA, USA, May 2012.
- [30] MULLEN, T. *Prototyping Augmented Reality*. Serious skills. Wiley, 2011.
- [31] NIEMEYER, G., PREUSCHE, C., Y HIRZINGER, G. Telerobotics. In *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 741-757.
- [32] NORMAN, D. A. The next ui breakthrough, part 2: physicality. *Interactions* 14 (July 2007), 46-47.
- [33] NORMAN, D. A. Natural user interfaces are not natural. *Interactions* 17 (May 2010), 6-10.
- [34] NORMAN, D. A., Y NIELSEN, J. Gestural interfaces: a step backward in usability. *Interactions* 17 (September 2010), 46-49.
- [35] PISKORSKI, S., BRULEZ, N., ELINE, P., Y D'HAeyer, F. Ar.drone developer guide. Parrot, December 2012. Revision: SDK 2.0.
- [36] POUNDS, P., MAHONY, R., HYNES, P., Y ROBERTS, J. Design of a four-rotor aerial robot. In *Proceedings of the 2002 Australasian Conference on Robotics and Automation* (Auckland, New Zealand, November 2002), Australasian Conference on Robotics and Automation (ACRA, 2002), pp. 145-150.

- [37] REAS, C., Y FRY, B. *Getting Started with Processing*, 1st ed. Make Books - Imprint of: O'Reilly Media, Sebastopol, CA, 2010.
- [38] REED, T., GEIS, J., Y DIETRICH, S. Skynet: a 3g-enabled mobile attack drone and stealth botmaster. In *Proceedings of the 5th USENIX conference on Offensive technologies* (Berkeley, CA, USA, 2011), WOOT'11, USENIX Association, pp. 4-4.
- [39] ROSS, S., BARKHUDAROV, N. M., SHANKAR, K. S., WENDEL, A., DEY, D., BAGNELL, J. A., Y HEBERT, M. Learning monocular reactive uav control in cluttered natural environments. In *arXiv:1211.1690v1* (November 2012).
- [40] SHARKEY, N. Automating warfare: Lessons learned from the drones. *Journal of Law, Information and Science* 21, 2 (2011/2012), 140-154.
- [41] SHARKEY, N. The evitability of autonomous robot warfare. *International Review of the Red Cross* 94, 886 (2012), 787-799.
- [42] SONNTAG, D. A study of quadrotor modelling. Master's thesis, Linköping University, Department of Computer and Information Science, Linköping, Sweden, March 2011.
- [43] TWU, P., YOUNG, R., CROIX, J.-P. D. L., EGERSTEDT, M., CHIPALKATTY, R., RAHMANI, A., RAMACHANDRAN, T., Y SHIVELY, J. *A Hardware Testbed for Multi-UAV Collaborative Ground Convoy Protection in Dynamic Environments*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Portland, OR, USA, Aug 2011, ch. AIAA Modeling and Simulation Technologies Conference.
- [44] ULRICH, K. T. *Design: Creation of Artifacts in Society*. Trustees of the University of Pennsylvania, 2011.
- [45] WEIBEL, R. E. Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system. Master's thesis, Massachusetts Institute of

Technology. Dept. of Aeronautics and Astronautics, Massachusetts, MA, USA, June 2005.

[46] WINGRAVE, C. A., WILLIAMSON, B., VARCHOLIK, P. D., ROSE, J., MILLER, A., CHARBONNEAU, E., BOTT, J., Y LAVIOLA JR., J. J. The wiimote and beyond: Spatially convenient devices for 3d user interfaces. *Computer Graphics and Applications, IEEE 30*, 2 (march-april 2010), 71-85.

[47] YOSHIDA, S., NARUMI, T., HASHIMOTO, S., TANIKAWA, T., INAMI, M., IGARASHI, T., Y HIROSE, M. Augmentation of the embodied cognition using a gesture-controlled flying robot. In *Interaction 2012* (Koto-ku, Tokyo, Japan, 2012), pp. 403–408. (in Japanese).

VII - ANEXOS

Anexo 1. Glosario

Acelerómetro: Sensor que mide la aceleración.

Giroscopio: Sensor que mide la velocidad de rotación.

Sketch: término que se emplea para significar un programa elaborado en el lenguaje Processing o en Arduino C.

Anexo 2. Post-Mortem

1. Metodología propuesta vs. Metodología realmente utilizada.

La metodología propuesta fue el **modelo unificado de diseño**, de Karl T. Ulrich [44] y esta fue la que realmente se utilizó.

2. Actividades propuestas vs. Actividades realizadas.

Las actividades propuestas tanto para la fase del ciclo de diseño, como para la fase de producción, fueron exactamente las mismas actividades realizadas.

3. Efectividad en la estimación de tiempos del proyecto

En la Ilustración 32 aparece el diagrama PERT/CPM propuesto.

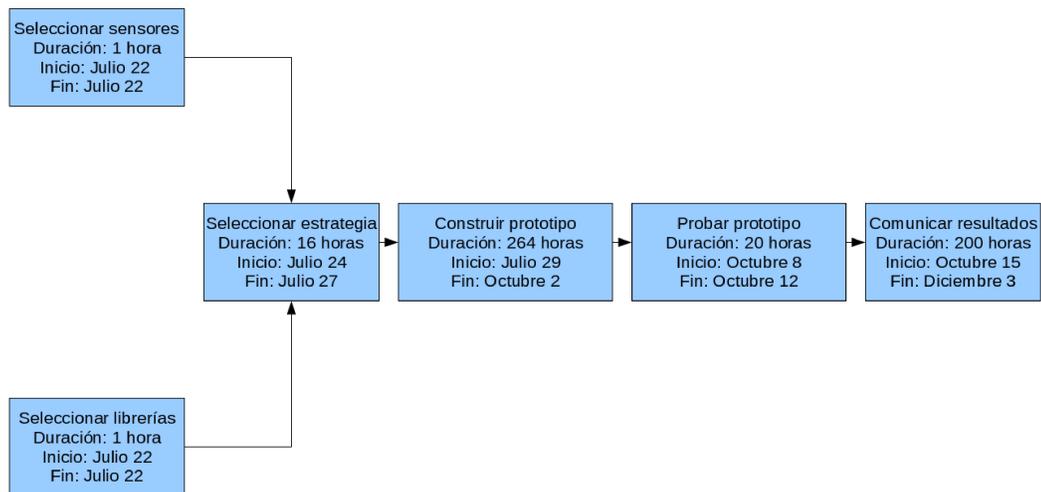


Ilustración 32.Diagrama PERT/CPM propuesto.

En la Tabla 3 se muestra el cronograma detallado con las horas de trabajo diario propuesto.

Actividad	Días	Horas por cada día
Seleccionar sensores y librerías	Julio 22	1 y 1 respectivamente
Holgura de tiempo	Julio 23	Holgura de tiempo
Seleccionar estrategia	Julio 24-27	4
Holgura de tiempo	Julio 28	Holgura de tiempo
Construir prototipo	Julio 29-Octubre 2	4
Holgura de tiempo	Octubre 3-7	Holgura de tiempo
Probar prototipo	Octubre 8-12	4
Holgura de tiempo	Octubre 13-14	Holgura de tiempo
Comunicar resultados	Octubre 15-Diciembre 3	4
Holgura de tiempo	Diciembre 4-10	Holgura de tiempo

Tabla 3. Cronograma detallado con las horas de trabajo diario propuesto.

En la Ilustración 33 se puede observar el diagrama PERT/CPM ajustado al principio del semestre.

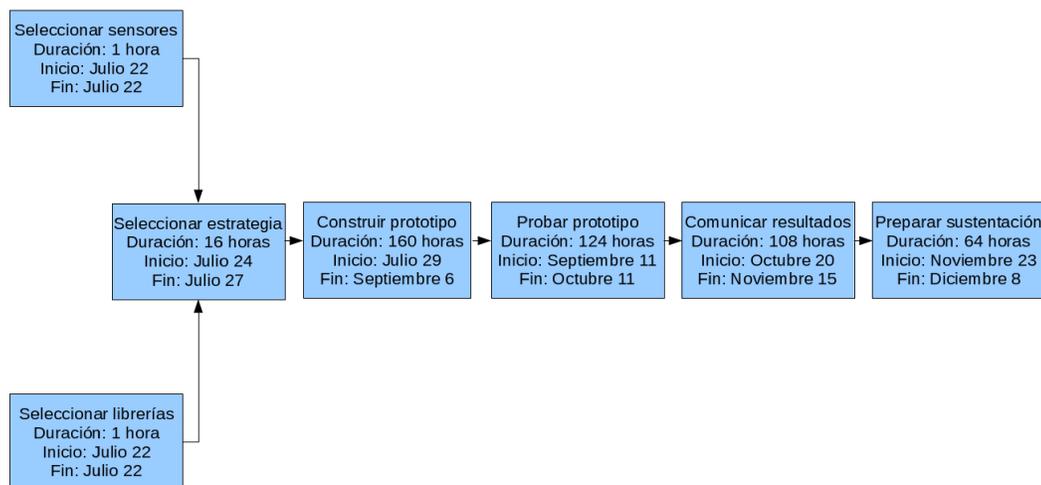


Ilustración 33. Diagrama PERT/CPM ajustado al principio del semestre.

En la Tabla 4 se presenta el cronograma detallado con las horas de trabajo diario ajustado al principio del semestre.

Actividad	Días	Horas por cada día
Seleccionar sensores y librerías	Julio 22	1 y 1 respectivamente
Holgura de tiempo	Julio 23	Holgura de tiempo
Seleccionar estrategia	Julio 24-27	4
Holgura de tiempo	Julio 28	Holgura de tiempo
Construir prototipo	Julio 29-Septiembre 6	4
Holgura de tiempo	Septiembre 7-10	Holgura de tiempo
Probar prototipo	Septiembre 11-Octubre 11	4
Holgura de tiempo	Octubre 12-19	Holgura de tiempo
Comunicar resultados	Octubre 20-Noviembre 15	4
Holgura de tiempo	Noviembre 16-22	Holgura de tiempo
Preparar sustentación	Noviembre 23-Diciembre 8	4

Tabla 4. Cronograma detallado con las horas de trabajo diario ajustado al principio del semestre.

La variación entre el cronograma propuesto y el ajustado a principio de semestre se debió a que cuando se hizo la propuesta se desconocía la fecha precisa de las entregas y de la sustentación. Se solicitó esta información el día 19 de Julio de 2013 a la Coordinación de Trabajos de Grado y se obtuvo el día 31 de Julio de 2013. Luego de realizar esta averiguación se creó una nueva actividad denominada preparar sustentación con 64 horas disponibles y se modificaron las siguientes actividades: construir prototipo se redujo a un 60.6% de lo propuesto, probar prototipo se incrementó a un 620% de lo propuesto y comunicar resultados se redujo a un 54% de lo propuesto.

4. Costo estimado vs. Costo real del proyecto

En la Tabla 5 se expone el costo estimado del trabajo de grado.

Recurso	Costo
Parrot AR.Drone 1.0 y dos baterías extra	\$ 1.200.000
Control remoto Wii™	\$ 139.000
Portátil	\$ 3.333.000
Otros items	\$ 600.000
TOTAL	\$ 5.272.000

Tabla 5.Costo estimado del trabajo de grado.

En la Tabla 6 se exhibe el costo real del trabajo de grado.

Recurso	Costo
Parrot AR.Drone 1.0 y dos baterías extra	\$ 1.200.000
Control remoto Wii™	\$ 139.000
Portátil	\$ 3.333.000
DFRobotShop Rover V2 (Xbee® Kit)	\$ 600.000
TOTAL	\$ 5.272.000

Tabla 6.Costo real del trabajo de grado.

5. Efectividad en la estimación y mitigación de los riesgos del proyecto.

En la Tabla 7 se indican los riesgos estimados para el trabajo de grado.

Riesgo	Técnica para minimizar el riesgo
Daño del control remoto Wii™	Tener un control de repuesto
Daño del Parrot AR.Drone 1.0	Tener un proveedor de partes de reemplazo
Tormenta geomagnética	Conocer con anticipación los días y las horas de la tormenta

Tabla 7. Riesgos estimados para el trabajo de grado.

A medida que avanza el ciclo solar 24, que comenzó en el año 1999, se espera que el número de tormentas geomagnéticas aumenten debido al cambio en el clima espacial y a la cercanía del máximo solar. Las tormentas geomagnéticas extremas (escala NOAA G5) pueden acabar con el suministro de energía eléctrica debido a que ciertas corrientes inducidas geomagnéticamente queman los transformadores de energía y hacen imposible las comunicaciones cableadas e inalámbricas (Wi-Fi®, Bluetooth®, GPS, enlaces satelitales, radio y televisión análoga y digital etc.). Algunas predicciones afirman que el máximo solar se alcanzará en algún punto comprendido entre los años 2013 y 2014 aunque se desconoce la potencia que tendrá el mismo evento [2].

Si se presentan tormentas de niveles G1 (menor), G2 (moderado), G3 (fuerte) es posible administrar el riesgo porque los satélites que vigilan el sol pueden advertir el problema e informar a la tierra sobre el mismo. El nivel G4 (severo) comienza a ser difícil de administrar porque los satélites van a presentar saltos de bits en sus CPUs debidos a la gran cantidad de energía a la que van a estar expuestos. El nivel G5 (extremo) no se puede manejar porque lo más probable es que los satélites se dañen y se funda el silicio de sus paneles solares dejándolos averiados y sin energía.

Afortunadamente, durante el desarrollo de este trabajo de grado se presentaron tormentas geomagnéticas suaves (niveles G1 y G2).

El único daño físico que se presentó fueron algunas fracturas en el casco protector del dron que se repararon soldando la parte afectada con una pistola de silicona caliente.

Anexo 3. Marcas usadas en este trabajo

Los nombres de las empresas o los productos mencionados en este escrito pueden ser marcas o marcas registradas por sus propietarios y son usados solamente para identificar y explicar algo en este texto, sin ninguna intención de infringir sus derechos.

Processing es un proyecto abierto iniciado por Ben Fry y Casey Reas.

Copyright © 2004-2013 Ben Fry y Casey Reas.

Copyright © 2001-2004 Instituto Tecnológico de Massachusetts.

Arduino es un proyecto abierto escrito, depurado y apoyado por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis.

Parrot es una marca registrada por PARROT SA.

AR.Drone es una marca de Parrot SA.

Wii™ es una marca registrada por la Sociedad Anónima Nintendo.

Bluetooth® es una marca registrada por el Grupo de Interés Especial de Bluetooth.

Wi-Fi® es una marca registrada por la Alianza Wi-Fi.

iPhone®, iPad®, iPod touch® son marcas registradas por Apple, Inc.

Android™ es una marca de Google Inc.

LabVIEW™ es una marca de National Instruments Corp.

Debian es una marca registrada por Software in the Public Interest, Inc.

Xbee® es una marca registrada por Digi International, Inc.

Linux® es una marca registrada por Linus Torvalds.

Slackware® es una marca registrada por Patrick Volkerding y Slackware Linux, Inc.

CARTA DE AUTORIZACIÓN DEL AUTOR

(Licencia de uso)

Bogotá, D.C., 31 de Enero de 2014.

Señores

Biblioteca Alfonso Borrero Cabal S.J.

Pontificia Universidad Javeriana

Ciudad

El suscrito:

Mauricio Armando González Rodríguez

, con C.C. No

79648234

En mi calidad de autor exclusivo de la obra titulada:

Seguimiento y realidad aumentada con un dron teleoperado

(por favor señale con una "x" las opciones que apliquen)

Tesis doctoral Trabajo de grado Premio o distinción: **Si** **No**

presentado y aprobado en el año **2013**, por medio del presente escrito autorizo

a la Pontificia Universidad Javeriana para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mi obra las atribuciones que se indican a continuación, teniendo en cuenta que en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autorizan a la Pontificia Universidad Javeriana, a los usuarios de la Biblioteca Alfonso Borrero Cabal S.J., así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado un convenio, son:

AUTORIZO	SI	NO
1. La conservación de los ejemplares necesarios en la sala de tesis y trabajos de grado de la Biblioteca.	x	
2. La consulta física (sólo en las instalaciones de la Biblioteca)	x	
3. La consulta electrónica - on line (a través del catálogo Biblos y el Repositorio Institucional)	x	
4. La reproducción por cualquier formato conocido o por conocer	x	
5. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet	x	
6. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previo convenio perfeccionado con la Pontificia Universidad Javeriana para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones	x	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

De manera complementaria, garantizo en mi calidad de estudiante y por ende autor exclusivo, que la Tesis o Trabajo de Grado en cuestión, es producto de mi plena autoría, de mi esfuerzo personal intelectual, como consecuencia de mi creación original particular y, por tanto, soy el único titular de la misma. Además, aseguro que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí competencia exclusiva, eximiendo de toda responsabilidad a la Pontificia Universidad Javeriana por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Pontificia Universidad Javeriana está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA:

Información Confidencial

Esta Tesis o Trabajo de Grado contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de una investigación que se adelanta y cuyos

resultados finales no se han publicado.

Si

No

NOMBRE COMPLETO	No. del documento de identidad	FIRMA
Mauricio Armando González Rodríguez	79648234	M. G.

Ingeniería

FACULTAD:

Ingeniería de Sistemas

PROGRAMA ACADÉMICO:

BIBLIOTECA ALFONSO BORRERO CABAL, S.J.

DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADO

FORMULARIO

TÍTULO COMPLETO DE LA TESIS DOCTORAL O TRABAJO DE GRADO			
Seguimiento y realidad aumentada con un dron teleoperado			
SUBTÍTULO, SI LO TIENE			
AUTOR O AUTORES			
Apellidos Completos		Nombres Completos	
González Rodríguez		Mauricio Armando	
DIRECTOR (ES) TESIS O DEL TRABAJO DE GRADO			
Apellidos Completos		Nombres Completos	
Cárdenas Herrera		Pedro Fabián	
FACULTAD			
Ingeniería			
PROGRAMA ACADÉMICO			
Tipo de programa (seleccione con "x")			
Pregrado	Especialización	Maestría	Doctorado
x			

Nombre del programa académico						
Ingeniería de Sistemas						
Nombres y apellidos del director del programa académico						
Germán Alberto Chavarro Flórez						
TRABAJO PARA OPTAR AL TÍTULO DE:						
Ingeniero de Sistemas						
PREMIO O DISTINCIÓN <i>(En caso de ser LAUREADAS o tener una mención especial):</i>						
CIUDAD		AÑO DE PRESENTACIÓN DE LA TESIS O DEL TRABAJO DE GRADO			NÚMERO DE PÁGINAS	
Bogotá		2013			78	
TIPO DE ILUSTRACIONES (seleccione con "x")						
Dibujos	Pinturas	Tablas, gráficos y diagramas	Planos	Mapas	Fotografías	Partituras
		x			x	
SOFTWARE REQUERIDO O ESPECIALIZADO PARA LA LECTURA DEL DOCUMENTO						
Nota: En caso de que el software (programa especializado requerido) no se encuentre licenciado por la Universidad a través de la Biblioteca (previa consulta al estudiante), el texto de la Tesis o Trabajo de Grado quedará solamente en formato PDF.						
MATERIAL ACOMPAÑANTE						
TIPO	DURACIÓN	CANTIDAD	FORMATO			

	(minutos)		CD	DVD	Otro ¿Cuál?
Vídeo					
Audio					
Multimedia					
Producción electrónica					
Otro ¿Cuál?	Sitio web	1			Archivos html, pdf, odt, bib, pde, ogg, ogv, jar, dat, hiro, txt, jpg y zip

DESCRIPTORES O PALABRAS CLAVE EN ESPAÑOL E INGLÉS

Son los términos que definen los temas que identifican el contenido. *(En caso de duda para designar estos descriptores, se recomienda consultar con la Sección de Desarrollo de Colecciones de la Biblioteca Alfonso Borrero Cabal S.J en el correo biblioteca@javeriana.edu.co, donde se les orientará).*

ESPAÑOL	INGLÉS
Dron, UAS, UAV, seguimiento,	Drone, UAS, UAV, tracking,
realidad aumentada, teleoperación,	augmented reality, teleoperation,
Parrot AR.Drone, rover,	Parrot AR.Drone, rover,
control remoto Wii, ARToolKit,	Wii remote control, ARToolKit,
marcador visual, Filtro Paso Bajo,	visual marker, Low Pass Filter,
controlador PID, Processing,	PID controller, Processing,

Arduino, cuadricóptero, cuadrirotor,	Arduino, quadcopter, quadrotor,
robot móvil, robot aéreo, robótica	mobile robot, aerial robot, robotics

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS

(Máximo 250 palabras - 1530 caracteres)

RESUMEN

En este escrito se desarrolló un prototipo de software que permite a un cuadrirotor Parrot AR.Drone 1.0 seguir a un DFRobotShop Rover V2 y agregar realidad aumentada. El enfoque que se empleó requiere un marcador visual ARToolKit colocado sobre el rover y usa la cámara incorporada en el dron, un Filtro Paso Bajo de primer orden para filtrar el ruido del altímetro del dron y unos controladores PID para generar los movimientos de cabeceo, alabeo y velocidad vertical que permiten alinear el centro de la imagen de la cámara del dron con el centro del marcador visual ARToolKit del rover.

ABSTRACT

In this writing it was developed a software prototype that enables a Parrot AR.Drone 1.0 quadrotor to track a DFRobotShop Rover V2 and add augmented reality. The approach used requires an ARToolKit visual marker placed on the rover and uses the camera built into the drone, a 1st order Low Pass Filter to filter drone's altimeter noise and some PID controllers to generate the movements of pitch, roll and vertical speed to align the center of the drone's camera image with the center of the rover's ARToolKit visual marker.