SISTEMA VISUAL CON REFERENCIA DE APRILTAGS PARA NAVEGACIÓN DE UN DRONE

Ing. Francisco Javier Campos Navarro

***Resumen*— El objetivo principal de este proyecto es integrar varias tecnologías en un drone para crear un sistema de navegación autónomo, a través de un algoritmo y sistema de visión computacional abordo, el cual procesa imágenes y localiza las etiquetas (April Tag), mismas que proporcionan información para la navegación del drone en un ambiente cerrado.**

***Palabras clave*— Drone, vuelo autónomo, visión por computadora, detención de AprilTag.**

Introducción

Este documento pretende revisar las ventajas que brinda un sistema de orientación por AprilTag para la navegación de un drone en interiores; los sistemas dinámicos que requieran orientación, navegación y control se pueden ver fácilmente beneficiados con una integración de INS/GPS [1], sistema navegación inercial (INS) y del sistema de posicionamiento global (GPS), las ventajas de ambas tecnologías se combinan para ofrecer una solución de navegación completa continua, de alto ancho de banda, con una alta precisión a largo y corto plazo. La integración es adecuada para aplicaciones de navegación, tales como barcos o aviones.

En el caso de los drones que vuelan en interiores tienen dificultades para navegar y ubicar su posición en el transcurso de su movimiento, si cuenta con GPS la señal en interiores es intermitente y puede llegar a ser nula, si se contara con otros sistemas tales como: señales de radio frecuencia o un laboratorio con cámaras de capturas de movimiento para su navegación o ubicación no dota al drone de ser un ente independiente y le resta autonomía; en el caso de la investigación se desea un sistema para interiores donde el sistema dinámico en este caso un drone esté dotado de cierta autonomía con varios sensores ópticos para la medición de distancia respecto al suelo y visión computacional, con pocas referencias externas para su navegación autónoma en este caso AprilTags.

Descripción del Método

Configuración física del dispositivo

El proyecto se basa en un UAV (Unmanned Aerial Vehicle) vehículos aéreos no tripulados o actualmente denominado drone, este es un cuadricóptero instrumentado capaz de realizar vuelos de forma autónoma para mantener la altura y búsqueda de AprilTags por cuestiones de seguridad, el drone también puede ser controlado por radio control por si el vuelo autónomo saliera de control. El sistema está conformado por 4 motores brushless, 4 (ESC) Electronic Speed Controller que son los controladores de velocidad para que giren los motores a la velocidad deseada según el controlador de vuelo, un módulo de telemetría, un módulo para comunicación con el control remoto, un sensor óptico TF Mini LiDAR para medir distancia, este sensor nos proporciona la altura del drone respecto al suelo, un sensor OpenMV el cual será el módulo de visión, un Raspberry Pi 3 el cual se utiliza como computadora abordo para realizar tareas computacionales complejas que se requieran en pedido de tiempo corto y un controlador de vuelo PixHawk Racer R15. En la figura 1 se puede ver la intención entre los módulos del sistema en la figura 2 se puede ver el drone ensamblado con todos los sistemas.

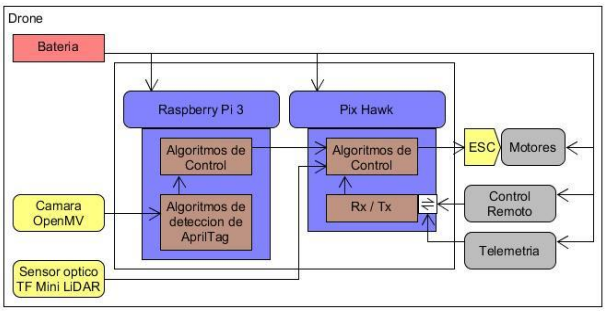

Figura 1. Diagrama Hardware, Software drone.

Figura 2. Integración del Drone

Descripción de lógica del dispositivo

El Raspberry Pi y el PixHawk Racer están conectados físicamente por comunicación serial como se observa en la figura 3, vía DroneKit-Python este a su vez utiliza el protocolo MAVLink. El Raspberry Pi 3 conectado vía USB a la cámara OpenMV; capturan el trayecto del drone en búsqueda de la AprilTag, un tipo de código de barras, pero en dos dimensiones, si la AprilTag no es localizada por el drone efectuara un aterrizaje después de haber avanzado 3 metros desde el punto de donde despego, si la AprilTag es localizada el drone debe centrarse a ella y aterrizar sobre la AprilTag.

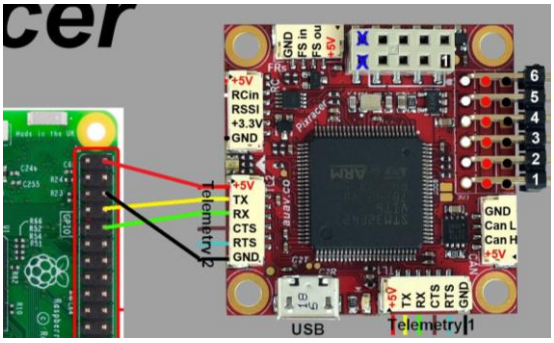

Figura 3. Raspberry Pi 3 conexión con PixHawk Racer.

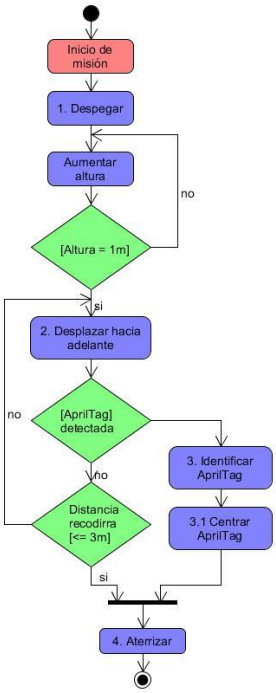

Figura 4. Diagrama de la lógica de ejecución.

Objetivos

El objetivo de este proyecto es usar un drone instrumentado para completar una misión principal o interrumpir y ejecutar la misión secundaria, los pasos para realizar las misiones se separaron en tareas que son las siguientes:

Misión principal

1. Despegue del drone a una altura de 1 metro.
2. Desplazarse en línea recta hacia el frente a una distancia cercada a 3 metros con altura constante de 1 metro.
3. Aterrizar drone.

Misión secundaria

1. Despegue del drone a una altura de 1 metro.
2. Desplazarse en línea recta hacia el frente a una distancia cercada a 3 metros con altura constante de 1 metro.
3. Identificar el AprilTags.

3.1 Centrar respecto AprilTag.

4. Aterrizar drone.

En la figura 4 se puede observar el diagrama de la lógica de ejecución del algoritmo tanto de la misión principal como secundaria.

Hardware, Software y Conceptos importantes

Los componentes principales del hardware del drone son el PixHawk Racer 15, Raspberry Pi 3, OpenMV, AprilTag a continuación se dará una breve explicación de cada uno de ellos y su interacción.

PixHawk Racer 15 es una tarjeta controladora de vuelo basada en el software de ArduPilot creada para propósito general pero orientada para los drones pequeños o carreras, la tarjeta tiene varios sensores como acelerómetro, giroscopio, magnetómetro y barómetro, también cuenta con varios puertos de salida y entrada como, 2 puertos de telemetría, puertos UARTs, puerto CAN, puerto USB, puertos para el radio control, puerto I2C, varios puerto más y el control de los ESC que controlan los motores del drone, el segundo puerto de telemetría será

utilizado para la comunicación serial con el Raspberry Pi 3 a través del GPIO pins 1, 3, 4 y 5 como se observa en la figura 3.

El Raspberry Pi 3 es un ordenador de placa de reducidas dimensiones con un sistema operativo con base en Linux en este caso Debian, el cual permite ejecutar programas por medio de la comunicación serial y protocolos MAVLink interactuar con la tarjeta controladora del drone, a través de la conexión serial; También cuenta con (SSH) Secure Shell por el cual se puede tener acceso a la terminal para ejecutar varias aplicaciones como DroneKit-Python y otras librerías de Python para poder cargar el algoritmo y ejecutar scripts.

DroneKit-Python es un proyecto de código abierto que es compatible con vehículos que se comunican mediante el protocolo MAVLink, pensado para crear aplicaciones que se ejecutan en una computadora complementaria integrada al sistema para comunicarse con el controlador de vuelo, las aplicaciones que se pueden desarrollar son variadas y mejoran significativamente el poder de computo del controlador de vuelo agregando mayor inteligencia computacional al vehículo y mayor rango de tareas a realizar, también proporciona acceso a la información de telemetría, estado y parámetros de un vuelo, permitiendo tanto la gestión de la misión como el control directo sobre el movimiento y las operaciones del vehículo.

El sensor OpenMV Cam M7 es un módulo de visión de bajo costo programable con Python pensado para crear algoritmos de visión computacional en una tarjeta como Arduino, pero con una cámara de bajo consumo con un procesador STM32F765VI ARM Cortex M7 con entradas y salidas de 3.3V a 5V, con un sensor OV7725 capas de tomar 640x480 8bits imágenes en escala de grises o 640x480 16 bits RGB565 imágenes o 60 FPS a una resolución de 320x240. El modulo puede ser aplicado con algunas funciones como, detención de cara, seguimiento de ojos y detención donde la pupila está mirando, seguimiento de colores, seguimiento de línea, detención y decodificación de códigos QR y seguimiento de AprilTag detención de su rotación y tamaño.

Figura 5. OpenMV.

AprilTag [2] es un código de barras bidimensional cuadrado que almacena datos codificados, es la evolución del código de barras y de los códigos QR, es un módulo para almacenar información en una matriz de puntos pero con menor cantidad de información (entre 4 y 12 bits), esto hace que sean detectadas con mayor robustez siendo usadas en una gran variedad de tareas como realidad aumentada o robótica y calibración de cámaras, las etiquetas puede ser impresas, las AprilTag son detectadas por software de detención computacional con gran precisión de la posición en tres dimensiones así como su orientación y se puede identificar la posición relativa de la cámara. Las librerías de las AprilTag están implementadas en C sin dependencias externas, están diseñadas para incluirlas en otras aplicaciones, fácilmente portables en dispositivos embebidos y para un performance en tiempo real. Existen varias familias de AprilTag, el nombre 36h11 indica que estas son etiquetas de 36 bits (matriz 6x6) con una distancia de Hamming mínima de 11 bits entre códigos, son 518 etiquetas de esta familia.

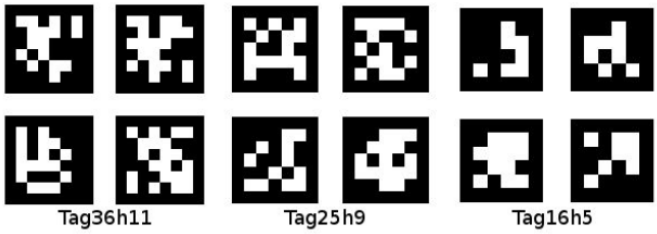

Figura 6. Familias de AprilTags.

Cómo funciona el algoritmo de AprilTag

El algoritmo consta de nueve pasos:

1. Convierte la imagen a escala de grises de punto flotante (valores de píxel entre 0.0 y 1.0) y aplique un desenfoque Gaussiano.
2. Calcula el gradiente local (magnitud y dirección) en cada píxel.
3. Genera una lista de bordes, agrupando píxeles conectados con direcciones similares. Un borde está presente si la magnitud del gradiente para ambos píxeles está significativamente por encima de cero.
4. Crear grupos desde los bordes.

5. Pasar sobre los grupos, encajando líneas llamadas Segmentos.
6. Para cada segmento, busque segmentos que comiencen donde termina este segmento.
7. Buscar todos los segmentos conectados para encontrar bucles de longitud 4, llamados Quads. Cada quad representa el borde negro alrededor de una etiqueta candidata.
8. Decodificar los quads mirando los píxeles dentro del borde para ver si representan un código de etiqueta válido y generar una lista de TagDetections
9. Buscar las TagDetections superpuestas y tome las mejores (la distancia de Hamming más baja o el perímetro más grande); desecha el resto.

Avances

Diseño del sistema: Pasos próximos a ser implementados

Tomando en cuenta que el Raspberry Pi es el dispositivo con mayor capacidad de computo se establece como el dispositivo maestro y los demás sistemas serán nodos que brindan información. Los componentes hardware y la interacción con el software así como la relación de control es la siguiente: el control del ArduPilot que trabaja internamente en el PixHawk Racer hacer el control del vuelo mantener el drone en el aire, y controlar los motores para hacer los movimientos (roll, pitch, y yaw) cabeceo, inclinación y guiñada; el software de DroneKit-Python en el Raspberry Pi corre scripts donde solicita información al OpenMV de los AprilTags detectados en Python y mantiene comunicación con el PixHawk por medio de protocolo MAVLink con una comunicación serial, para mandar comandos de movimiento hacia los motores.

En la figura 7 se muestra un avance del algoritmo en el IDE del OpenMV conectada a la Raspberry Pi por medio de USB, corriendo un programa para detectar la AprilTags, en el programa se detectan las AprilTags y se crea un cuadrado rojo para enmarcar la etiqueta detectada con una cruz verde en el centro del cuadrado y líneas amarillas que se dibujan en toda la imagen para ver hacia que cuadrante esta el centro del AprilTag y saber su ubicación relativa siendo el centro de la imagen el posición X = 0.0, Y = 0.0, con estos valores el drone tendrá una referencia y saber hacia que lado debe moverse para central el AprilTag.

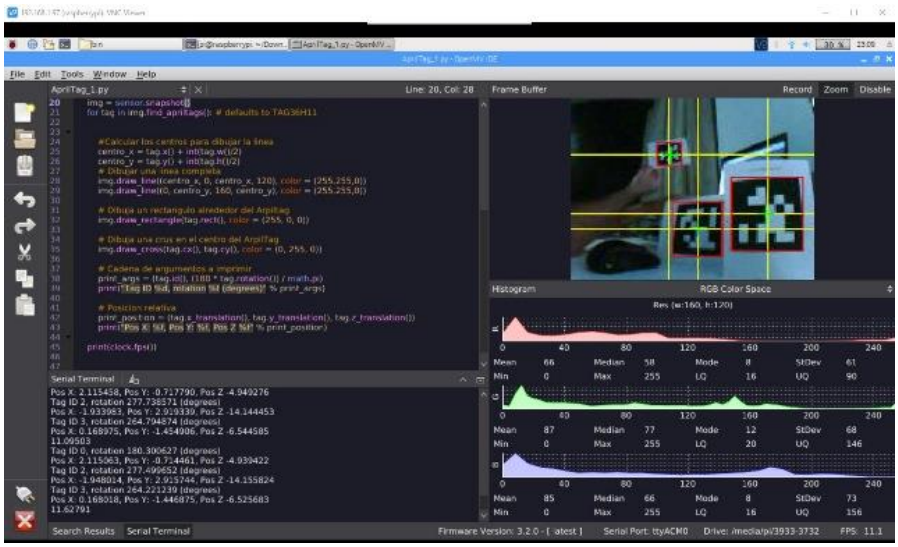

Figura 7. IDE OpenMV corriendo programa para detección de AprilTags

Descripción técnica de todos los elementos del sistema trabajando en conjunto. (caso misión secundaria con AprilTag detectada)

1. Establecer comunicación PixHawk con Raspberry Pi a través de Dronekit Python y protocolo MAVLink MAVProxy.
2. Armar el drone y realizar despegue, mantener altura y posición del drone.
3. Desplazarse hacia adelante, mantener altura y posición del drone.
4. Solicitar información del OpenMV acerca de la ubicación del AprilTag a través de Dronekit Python.
5. Ajustar el drone en dirección a los AprilTag.
6. Centrar el sistema y avanzar en dirección a los AprilTag.

7. Aterrizar próximo a la AprilTag.

Comentarios Finales

Aun no se cuenta con resultados experimentales el proyecto aún está en pruebas de unidad, aún no trabajan en conjunto todos los sistemas, faltan pruebas de los elementos independientes, partiendo de un drone con un vuelo estable y con los sensores ya montados y probados, la implementación del control debería ser sencilla, uno de los objetivo es entender la lógica del drone y como poder manipularlo desde otros sistemas, en este caso el Raspberry Pi y con Dronekit se realiza esta interacción con ArduPilot y MAVLink.

Se planteó el avance por etapas, la primera etapa consistió en la creación del drone, la siguiente etapa es implementar sensores para el vuelo estable y control de altura, en paralelo se trabaja en la etapa de pruebas con el OpenMV y la detención de AprilTags la penúltima etapa será montar todo en un sistema buscar la forma de que el sistema no sea muy pesado y la última etapa será realizar pruebas de integración vuelos y validación de los objetivos.

Sin la amplia variedad de librerías y ejemplos de Dronekit, así como de OpenMV la implementación de este proyecto sería mucho más complejo; Se busco que el lenguaje Python fuera el factor en común en todos los dispositivos para restar complejidad al unir los elementos en el sistema final.

Referencias

[1] España, Martin (2016). “Sistemas de Navegación Integrada con Aplicaciones” CONAE Comisión Nacional de Actividades Espaciales.

[2] E. Olsen, “AprilTag.” (en línea) Consultado por Internet el 02 de abril de 201. Dirección de internet: https://april.eecs.umich.edu/software/apriltag.html.

[3] Jeff Deeds, Zach Engstrom, Caleb Gill, and Zack Woods “Vision Based Autonomous Control of a Quadcopter” Department of Electrical and Computer Engineering, Bradley University, May 2, 2017