



| | | | |
|--|-------------|----------------|--|
| Apellidos, nome / Apellidos, nombre: Sánchez Canzobre, David | DNI: | e-mail: | Teléfono de contacto: |
| Título: Control de un cuatrirrotor con un miniPC | | | Mención cursada: Enxeñaría de Computadores |

Resumen / Resumo:

En este Trabajo de Fin de Grado controlamos de forma autónoma un cuatrirrotor (o dron) desde un miniPC adherido a él. En el miniPC se hace todo el procesado de los datos recogidos por los sensores del cuatrirrotor y en base a esos datos decide que acción va a realizar a continuación. El hecho de que el control se haga abordo del cuatrirrotor elimina la restrcción de la distancia a la que se encuentra el dron de la estación base donde se encuentra el control.

La principal aplicación de este TFG es el uso del dron en tareas de monitorización de zonas que serían difíciles de vigilar de otro modo, ya sea por lo arriesgado del lugar (por ejemplo monitorizar una zona catastrófica) o por los costes económicos de desplegar una solución más tradicional (por ejemplo si quisiéramos tener vigilada una gran superficie sería más económico tener un dron patrullando que cubrir toda esa gran superficie con cámaras).

El trabajo se ha dividido en dos grandes bloques. Por un lado la selección del hardware y el software, y por otro lado el diseño y la implementación de la arquitectura que se encarga del control autónomo del cuatrirrotor.

En cuanto al hardware hemos elegido como cuatrirrotor un Parrot AR Drone 2.0, como miniPC el Rikomagic MK802IV y la batería EasyAcc U-bright 3000 mAh Ultra. Para el control manual hemos seleccionado el mando Sony Dualshock 3. Por otra parte, el sistema operativo que hemos instalado en el miniPC es el Ubuntu 12.04 para ARM, el entorno de desarrollo de robótica ROS programado en Python y en C++ y el paquete de ROS ARDrone autonomy para comunicarnos con el dron.

Hemos diseñado una arquitectura de control independiente del hardware y del software que hemos elegido, lo que nos permite hacer cambios en una parte del sistema (tanto hardware como software) con un impacto mínimo. La arquitectura es modular, escalable y distribuida.

Por simplicidad, decidimos utilizar marcadores artificiales para identificar puntos en el entorno. Después de estudiar diversas alternativas, nos decantamos por los marcadores ArUco, ya que nos proporciona un gran número de identificadores diferentes, es muy robusto y rápido y nos da tanto distancia como orientación.

El cálculo de la ruta se realiza off-line y se carga desde un fichero que contiene los destinos (identificados con marcadores ArUco) y las órdenes necesarias para ir de un punto a otro (p.e., girar, colocarse a una determinada distancia del marcador, etc.).

Para implementar la arquitectura utilizamos varios nodos ROS (programas) que se comunican entre sí mediante paso de mensajes. Cada nodo se encarga de una tarea concreta: detectar los marcadores, centrar el dron respecto a un marcador, seleccionar la orden de control al dron, monitorizar el sistema, etc.

Hemos demostrado el funcionamiento de la detección de marcadores y para comprobar las capacidades del dron y del miniPC en condiciones normales de operación.

Por último, hemos realizado pruebas tanto de los nodos de control visual como de la ejecución de rutas simples.

Posibles aplicaciones / Posibles aplicacións:

- Vigilancia de grandes superficies comerciales. Pudiendo realizar el seguimiento de personas con un comportamiento sospechoso.
- Búsqueda y localización rápida de víctimas en zonas catastróficas.
- Monitorizar las condiciones (temperatura, humedad, etc) de grandes extensiones de cultivo, de forma rápida, con el fin de mejorar la productividad de las tierras.

Etapas para su desarrollo futuro / Etapas para o seu desenvolvemento futuro:

- 1º Portar nuestra arquitectura a un dron que tenga mayor capacidad de carga. Lo que nos permitirá utilizar mejores sensores (por ejemplo una cámara RGB-D) que nos permitan utilizar mejores soluciones para el mapeado y la localización de dron (por ejemplo Rtabmap).
- 2º Desarrollar un algoritmo de evitación de obstáculos, para realizar vuelos más seguros.
- 3º Desarrollar algoritmos específicos para cada aplicación: detección de personas, detección de caras para el seguimiento de una persona en concreto, etc.

Imágenes representativas:

