**Control del dron en espacios interiores**

La gran mayoría de las aplicaciones que forman parte del Drone Engineering Ecosystem son operativas en espacios exteriores con buena cobertura GPS, como es el caso del DronLab. En esa situación es posible desarrollar aplicaciones que dirijan al dron a puntos geo-localizados (lat, lon) o limitar sus movimientos con un geofence.

Sin embargo, no todo el mundo tiene un espacio cerrado con cobertura GPS. La cuestión es cómo conseguir que ese mismo tipo de aplicaciones (con las adaptaciones necesarias) puedan funcionar en espacios cerrados sin cobertura GPS, como puede ser el gimnasio de un centro educativo.

Esa es la cuestión a la que damos respuesta en este repositorio. Para ello, en las secciones siguientes haremos algunas consideraciones teóricas, veremos que dispositivos deberíamos instalar en el dron y cuáles son las configuraciones necesarias. Finalmente, proporcionamos una aplicación tipo Dashboard en Python que demuestra una variedad de opciones de control del dron, que puede ser la base de muchas y divertidas aplicaciones perfectamente operativas en espacios interiores sin GPS.

**Estabilidad y navegabilidad**

Para que podamos controlarlo desde nuestros programas el dron debe tener estabilidad y navegabilidad. El dron tiene estabilidad si se mantiene estable en su posición en el punto del espacio en el que lo hayamos situado. El dron tiene navegabilidad si es capaz de ejecutar correctamente operaciones para colocarse en un punto dado del espacio o volar en una dirección dada.

**Estabilidad**

Para mantenerse estable, el dron necesita información de sensores externos que le permitan estimar su posición xyz en el espacio, de manera que cualquier desviación indeseada respecto a esa posición (por ejemplo, por acción del viento) pueda ser detectada por el dron, que corregirá automáticamente su posición.

El autopiloto ya tiene un barómetro que le permite estimar la altura a la que se encuentra. Cuando volamos en modo AltHold, el dron es capaz de mantenerse estable en altura gracias a la información que le proporciona el barómetro. La precisión de la medida de la altura puede mejorarse si se instala un altímetro laser (ver apartado de configuraciones). Ambos dispositivos (barómetro y altímetro laser) pueden funcionar perfectamente en espacios interiores, en los que el dron puede volarse en modo AltHold, que permite una buena estabilidad en el eje z (es decir, en altura).

Para conseguir estabilidad en el plano xy el dron puede usar la señal GPS, que le permite estimar su posición en términos de latitud y longitud, y le permite detectar desviaciones indeseadas de esa posición de manera que puede corregirla inmediatamente. Cuando tenemos señal GPS podemos volar en modo Loiter, que permite mantener estable el dron en su posición xyz.

En espacios interiores puede ser que el dron no reciba bien la señal GPS, por lo que necesita otra fuente de información que le permita estimar su posición y mantener estabilidad en xyz. Esa información se la puede proporcionar un sensor de tipo Optical Flow. Se trata de una cámara que apunta al suelo y que permite detectar movimientos en el plano xy, puesto que un cambio en la imagen del suelo que detecta la cámara indica un desplazamiento del dron en ese plano. Naturalmente, para que el Optical Flow resulte útil es necesario que el suelo no sea uniforme, de manera que lo que se ve en un punto dado sea diferente de lo que se ve en los puntos cercanos. Por tanto, configurando adecuadamente un dispositivo Optival Flow (ver apartado de configuraciones) es posible armar el dron en modo Loiter y volar con estabilidad en las tres dimensiones xyz.

**Navegabilidad**

Para que el dron pueda navegar siguiendo las instrucciones de nuestros programas es necesario que disponga de un marco de referencia en el que pueda ubicarse (conocer su posición), ubicar cualquier otro punto dentro de ese marco de referencia (para volar hacia ese punto) y orientarse dentro del marco de referencia (por ejemplo, saber hacia dónde está el Norte).

La señal GPS permite al dron establecer un marco de referencia (que llamamos global) basado en las coordenadas geográficas (latitud, longitud y altura). Gracias a ese marco de referencia global, el dron puede ejecutar correctamente comandos como los que se muestran en la tabla 1 (que usan la librería pymavlink).

En el caso de un espacio interior al que no llega la señal GPS, el dron establece un marco de referencia local, apoyándose en la información que le proporciona el Optical Flow. Por una parte, el magnetómetro interno del autopiloto le permite saber dónde está el Norte, incluso en un espacio interior al que no llega la señal GPS. Por otra parte, al iniciarse el autopiloto establece como posición inicial (Home) la posición xyz = (0,0,0). La información que le proporciona el Optical Flow y el altímetro laser (o barómetro) permite ubicar cualquier posición xyz en ese marco local y volar hacia esa posición, o volar en una dirección determinada, si es instruido para ello. De hecho, de todos los comandos mostrados en la tabla 1, el único que no puede ejecutar es el primero, puesto que no tiene información sobre latitudes y longitudes.

|  |
| --- |
| vehicle.mav.send(  mavutil.mavlink.MAVLink\_set\_position\_target\_global\_int\_message(  10, vehicle.target\_system, vehicle.target\_component,  mavutil.mavlink.MAV\_FRAME\_GLOBAL\_RELATIVE\_ALT, int(0b110111111000),  int(lat \* 10 \*\* 7),int(lon \* 10 \*\* 7), alt,  0, 0, 0, 0, 0, 0, 0, 0) |
| El dron se dirige a la posición geográfica definida por (lat, lon, alt) en el marco de referencia global. |
| vehicle.mav.send(  mavutil.mavlink.MAVLink\_set\_position\_target\_local\_ned\_message(  10, vehicle.target\_system, vehicle.target\_component,  mavutil.mavlink.MAV\_FRAME\_LOCAL\_NED, # frame  0b110111111000,   x,  y,  z,  0, 0, 0, 0, 0, 0, 0, 0) |
| El dron se dirige a la posición xyz en el marco de referencia local, en el que el punto (0,0,0) es en ocupado por el dron en el momento de iniciar el autopiloto. El valor de x se entenderá como metros en dirección Norte (o Sur si x es negativo), el valor de y son metros en la dirección Este (u Oeste si y es negativo) y el valor de z se entiende como altitud en metros (atención porque valores negativos implican posiciones por encima de la altura 0). |
| vehicle.mav.send(  mavutil.mavlink.MAVLink\_set\_position\_target\_local\_ned\_message(  10, vehicle.target\_system, vehicle.target\_component,  mavutil.mavlink.MAV\_FRAME\_LOCAL\_OFFSET\_NED, # frame  0b110111111000,   x,  y,  z,  0, 0, 0, 0, 0, 0, 0, 0) |
| El dron se desplaza, respecto a la posición que ocupa, x metros hacia el Norte (o sur si negativo), y metros hacia el Este (u oeste si negativo) y z metros hacia abajo (o hacia arriba si es negativo). |
| vehicle.mav.send(  mavutil.mavlink.MAVLink\_set\_position\_target\_local\_ned\_message(  10, vehicle.target\_system, vehicle.target\_component,  mavutil.mavlink.MAV\_FRAME\_BODY\_OFFSET\_NED, # frame  0b110111111000,   x,  y,  z,  0, 0, 0, 0, 0, 0, 0, 0) |
| El dron se desplaza, respecto a la posición que ocupa, x metros hacia el delante (o atrás si negativo), y metros hacia la derecha (o izquierda si negativo) y z metros hacia abajo (o hacia arriba si es negativo). |

**Telemetría**

Normalmente, nuestros programas van a necesitar datos de telemetría (posición, heading, etc.) para facilitar al usuario la interacción con el dron (por ejemplo, para mostrar en un plano la posición que ocupa el dron en cada momento).

Para hacer que el dron nos envíe datos de telemetría es necesario ejecutar los comandos pymavlink siguiente:

frequency\_hz = 4 # enviará 4 paquetes de datos cada Segundo

# para recibir datos del marco global (por ejemplo, latitudes y longitudes)  
self.vehicle.mav.command\_long\_send(  
 self.vehicle.target\_system, self.vehicle.target\_component,  
 mavutil.mavlink.MAV\_CMD\_SET\_MESSAGE\_INTERVAL, 0,  
 mavutil.mavlink.MAVLINK\_MSG\_ID\_GLOBAL\_POSITION\_INT,   
 1e6 / frequency\_hz,  
 0, 0, 0, 0, # Unused parameters  
 0  
)  
# para recibir datos del marco local (por ejemplo, distancias recorridas respecto al

# home)

self.vehicle.mav.command\_long\_send(  
 self.vehicle.target\_system, self.vehicle.target\_component,  
 mavutil.mavlink.MAV\_CMD\_SET\_MESSAGE\_INTERVAL, 0,  
 mavutil.mavlink.MAVLINK\_MSG\_ID\_LOCAL\_POSITION\_NED, # The MAVLink message ID  
 1e6 / frequency\_hz,  
 0, 0, 0, 0, # Unused parameters  
 0

)

A partir de este momento podemos solicitar paquetes de datos de telemetría. Los que nos interesan más son los siguientes:

msg = vehicle.recv\_match(type='GLOBAL\_POSITION\_INT', blocking= True)  
msg = msg.to\_dict()  
lat = float(msg['lat'] / 10 \*\* 7)  
lon = float(msg['lon'] / 10 \*\* 7)  
alt = float(msg['relative\_alt']/1000)  
heading = float(msg['hdg'] / 100)

msg = vehicle.recv\_match(type='LOCAL\_POSITION\_NED', blocking=True)  
posX = msg.x # valores (metros) positivos hacia el Norte y negativos hacia el SUR  
posY= msg.y # valores (metros) positivos hacia el Este y negativos hacia el Oeste  
posZ= msg.z # valores (metros) negativos para posiciones por encima del origen

**Configuraciones necesarias**

Altímetro laser

La estimación de la altura puede mejorarse si se instala en el dron un altímetro laser, que es más preciso que el barómetro.

El altímetro laser que usamos en nuestras instalaciones es el que se describe aquí:

<https://ardupilot.org/copter/docs/common-lightware-lw20-lidar.html>

El altímetro laser debe conectarse al puerto IC2 del autopiloto. A continuación, deben ajustarse los valores de algunos parámetros, según indica la tabla siguiente:

|  |  |
| --- | --- |
| [RNGFND1\_TYPE](https://ardupilot.org/copter/docs/parameters.html#rngfnd1-type) | 7 (LightWareI2C) |
| [RNGFND1\_ADDR](https://ardupilot.org/copter/docs/parameters.html#rngfnd1-addr) | 102 (I2C Address of lidar in decimal) |
| [RNGFND1\_SCALING](https://ardupilot.org/copter/docs/parameters.html#rngfnd1-scaling) | 1 |
| [RNGFND1\_MIN\_CM](https://ardupilot.org/copter/docs/parameters.html#rngfnd1-min-cm) | 5 |
| [RNGFND1\_MAX\_CM](https://ardupilot.org/copter/docs/parameters.html#rngfnd1-max-cm) | 9500 **(***This is the distance in centimeters that the rangefinder can reliably read)* |
| [RNGFND1\_GNDCLEAR](https://ardupilot.org/copter/docs/parameters.html#rngfnd1-gndclear) | 15 *(Distance in centimetres from the range finder to the ground when the vehicle is landed. This value depends on how you have mounted the rangefinder)* |
| RNGFND1\_POS\_X | 0.119 *(Distance in centimetres from the range finder to the center of the autopilot, in dimensions XYZ. These values depend on how you have mounted the rangefinder)* |
| RNGFND1\_POS\_Y | 0.043 |
| RNGFND1\_POS\_Z | 0.064 |

|  |  |
| --- | --- |
| Si se ha realizado una correcta configuración, al conectar Mission Planner con el dron deberíamos poder leer los datos proporcionados por el altímetro en la pestaña Status, en el item sonargange. | ../_images/mp_rangefinder_lidarlite_testing.jpg |

Optical flow

El Optical Flow nos permite proporcionar al dron información de desplazamientos en el plano xy, en ausencia de señal GPS. El Optical flow que usamos en nuestras instalaciones es el descrito aquí:

<https://ardupilot.org/copter/docs/common-hereflow.html>

El Optical Flow debe conectarse al Puerto CAN2 de autopiloto. A continuación, deben ajustarse los valores de algunos parámetros, según indica la tabla siguiente:

|  |  |
| --- | --- |
| CAN\_P2\_DRIVER | 1 |
| FLOW\_TYPE | 6 (después de esto reiniciar el autopiloto para acceder a los parámetros restantes) |
| FLOW\_POS\_X | 0.119 *(Distance in centimetres from the optical flow to the center of the autopilot, in dimensions XYZ. These values depend on how you have mounted the optical flow)* |
| FLOW\_POS\_Y | 0 |
| FLOW\_POS\_Z | 0.064 |

|  |  |
| --- | --- |
| ../_images/PX4Flow_CheckForData_MP.png | Si se ha realizado una correcta configuración, al conectar Mission Planner con el dron deberíamos poder leer los datos proporcionados por el Optical flow en la pestanya Status |

Configuración del EKF (Extended Kalman Filter)

El autopiloto tiene implementados varios algoritmos (que denominamos EK2 y EK3) que usan el filtro de Kalman para estimar la posición, velocidad y orientación del dron. Esos algoritmos utilizan la información proporcionada por los sensores. Es necesario ahora indicar al autopiloto que algoritmo debe usar y qué sensores debe usar ese algoritmo.

Se recomienda utilizar el algoritmo EK3, que es más avanzado. Además, como se indica más adelante, el algoritmo EK2 ya no está disponible en las versiones más actuales del firmware del autopiloto. En cualquier caso, indicamos aquí como debe hacerse la configuración tanto del EK2 como del EK3.

La configuración para el EKF2 es la siguiente:

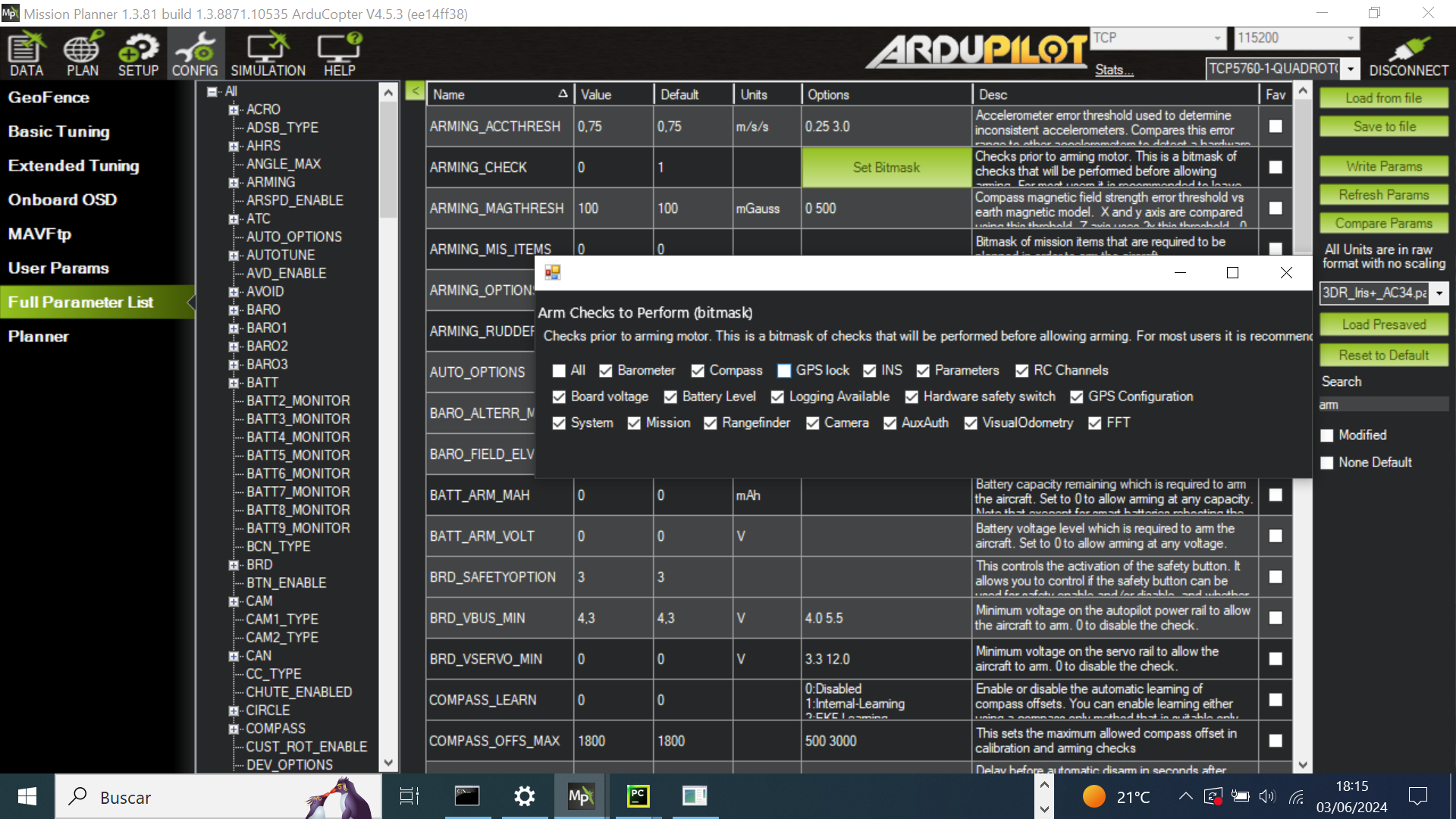
|  |  |
| --- | --- |
| AHRS\_EKF\_TYPE | 2 |
| EK2\_ENABLE | 1 (después hay que reiniciar el autopiloto para acceder al resto de parámetros) |
| EK2\_IMU\_MASK | 7 (este y los siguientes es para que use las tres IMUs) |
| INS\_USE | 1 |
| INS\_USE2 | 1 |
| INS\_USE3 | 1 |
| EK2\_ALT\_SOURCE | 1 (para que use el altímetro laser en vez del barómetro) |
| EK2\_GPS\_TYPE | 3 (Optical Flow) / 0 (GPS) |

Con todas estas configuraciones, para poder armar he tenido que deshabilitar todos los preAlarm checks. Pero seguro que esa deshabilitación puede hacerse de forma más selectiva. Hay que investigar eso.

Como se ha indicado antes, el algoritmo EKF2 ya no está operativo en las versiones más actuales del firmware del autopiloto (por ejemplo, ya no está en la versión 4.5.1). Para esas versiones más actuales debe usarse el EK3, cuya configuración se describe a continuación.

|  |  |
| --- | --- |
| AHRS\_EKF\_TYPE | 3 |
| EK3\_ENABLE | 1 (después hay que reiniciar el autopiloto para acceder al resto de parámetros) |
| [EK3\_SRC1\_POSXY](https://ardupilot.org/plane/docs/parameters.html#ek3-src1-posxy) | 0 (Optical Flow) / 3 (GPS) |
| [EK3\_SRC1\_VELXY](https://ardupilot.org/plane/docs/parameters.html#ek3-src1-velxy) | 5 (Optical Flow) / 3 (GPS) |
| [EK3\_SRC1\_POSZ](https://ardupilot.org/plane/docs/parameters.html#ek3-src1-posz) | 1 (Barómetro) / 2 (Altímetro laser) |
| [EK3\_SRC1\_VELZ](https://ardupilot.org/plane/docs/parameters.html#ek3-src1-velz) | 0 |
| [EK3\_SRC1\_YAW](https://ardupilot.org/plane/docs/parameters.html#ek3-src1-yaw) | 1 |
| [EK3\_SRC\_OPTIONS](https://ardupilot.org/plane/docs/parameters.html#ek3-src-options) | 0 |

En el caso de que se configura para usar el Optical Flow y no el GPS es necesario deshabilitar alguno de los pre-arm checks tal y como indica la figura:



Una vez realizadas todas estas configuraciones el dron debería armar en modo Loiter y deberíamos poder volarlo con buena estabilidad en interiores, usando la emisora de radio. Para ese escenario se recomienda tener configurado el modo Land para hacer que el dron aterrice inmediatamente en el punto que sobrevuela, el caso de que surja cualquier problema.