



Universitat d'Alacant
Universidad de Alicante



Grado en Ingeniería Robótica

Robots móviles

Proyecto teórico: Localización y control de un dron

Carmen Ballester Bernabeu

Carlos Castellanos Ormeño

Adrián Trujillo López

Índice

1. Introducción	2
2. Sistemas de control de drones quadrotor	2
2.1. Sistemas de control tradicional	3
2.1.1. Controladores lineales	4
2.1.2. Controladores no lineales	5
2.1.3. Controladores basados en aprendizaje	5
2.2. Sistemas de control y navegación experimentales	6
2.2.1. Usando visión por computador	6
2.2.2. Sistemas basados en GNSS	7
3. Sistema de detección	8
3.1. Códigos QR	8
3.1.1. Aplicaciones prácticas	9
3.2. Visión estereoscópica	11
4. Sistema de localización	12
4.1. Calibración	13
4.2. Triangulación	14
4.2.1. Matching	15
4.3. Transformaciones	15
5. Referencias	17

1. Introducción

El estudio de estrategias de localización de drones es un caso de investigación actualmente, debido a la complejidad de estimar una posición en un espacio tridimensional de un sistema que puede moverse libremente en los tres ejes del mismo. Esto, a su vez, debe hacerse en un tiempo de ejecución lo suficientemente rápido para que no se vea comprometido el control del dron y, por consiguiente, la realización de la tarea. Sin embargo, una buena implementación que resuelva estos inconvenientes permitiría que estos aparatos realizarán tareas tanto en entornos en zonas exteriores como en espacios interiores.

En el caso de este trabajo se investigará en una estrategia adaptada a espacios interiores, puesto que se busca utilizar códigos QR a modo de puntos de referencia del espacio 3D en el que debe operar el dron. En los siguientes apartados nos centraremos en la explicación de los procesos que deben llevarse a cabo en esta estrategia de localización, así como, el posterior control del movimiento del dron a raíz de los datos obtenidos. Pero primeramente se realizará una introducción a los diferentes tipos de control que existen y sobre los que se ha investigado para este proyecto.

Cabe mencionar que la ejecución de esta metodología se puede llevar a cabo de forma local en el propio dron, por lo que no es necesario establecer una conexión para obtener datos de elementos exteriores, al mismo tiempo, que se suprimen la necesidad de herramientas como el GPS, cuya posición estimada en espacios interiores resulta poco precisa.

2. Sistemas de control de drones quadrotor

Los drones quadrotores son los más famosos y utilizados actualmente para diferentes aplicaciones tanto militares como comerciales. Estos drones como se puede deducir por su nombre son los que tienen una configuración de 4 hélices con 4 motores como se puede ver en la figura 1.

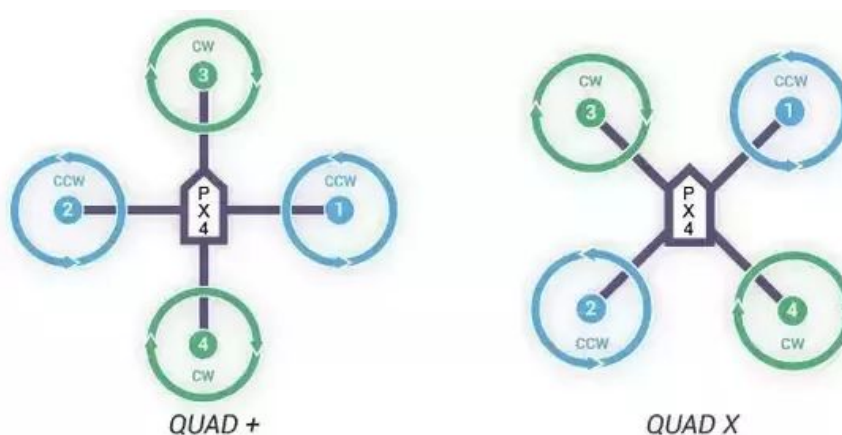


Figura 1: Configuración de los drones quadrotor.

Este tipo de drones se han ganado su popularidad gracias a su simple estructura, su capacidad para flotar en el aire y su alta maniobrabilidad para moverse en el espacio cartesiano. Además

se han utilizado en una gran cantidad de aplicaciones lo que ha permitido su desarrollo a una velocidad más rápida. Las aplicaciones en las que se suelen utilizar estos tipos de drones son por ejemplo : Vigilancia urbana, monitoreo ambiental, vigilancia en ambientes hostiles, misiones de rescates, grabación de películas, transportes y supervisión de construcciones civiles.

2.1. Sistemas de control tradicional

Los sistemas de control tradicionales son aquellos que se engloban en las últimas dos décadas estos se pueden diferenciar en tres tipos de control: linear, no linear y controladores inteligentes basados en aprendizaje. Donde en cada categoría se desarrollan varios métodos de control como se puede ver en la figura 2.

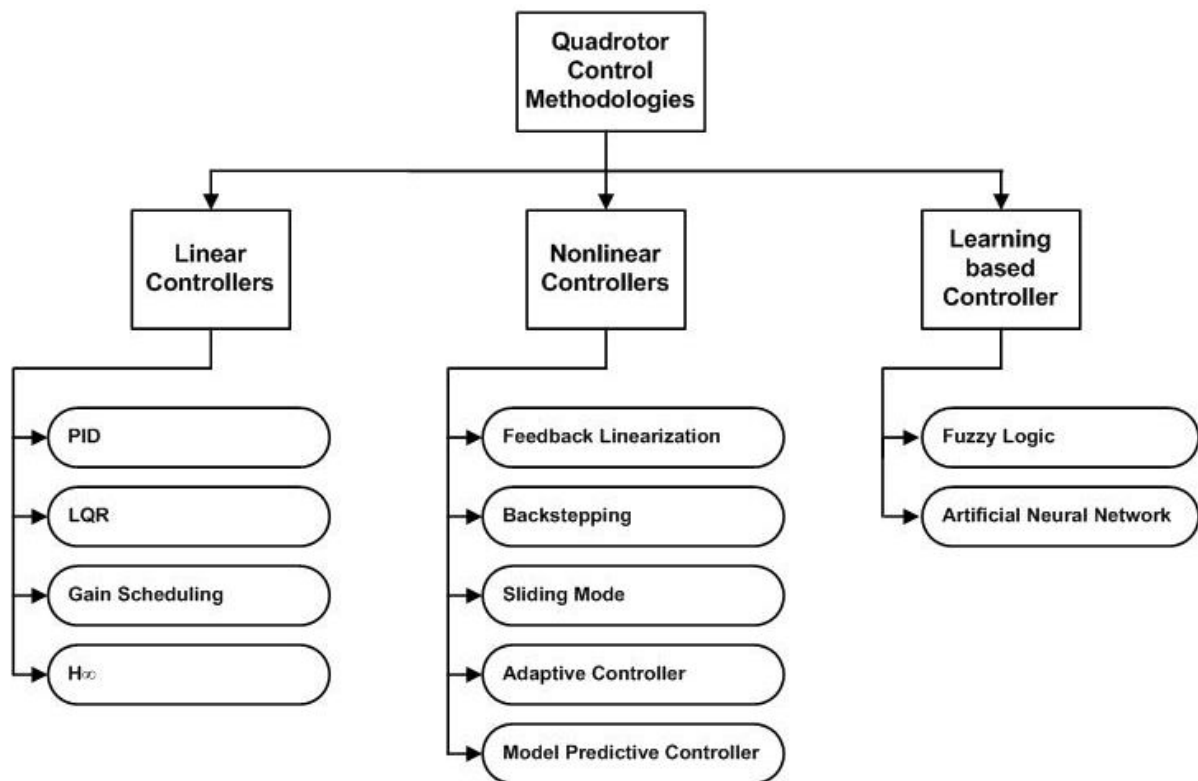


Figura 2: Esquema de los diferentes tipos de control para drones quadrotores.

Los quadrotores son sistemas subactuados inestables que necesitan de un controlador de bucle cerrado para poder funcionar de manera eficiente. La arquitectura se compone de dos bucles de control, uno es un bucle interno que se encargaría de controlar la orientación mientras que un bucle externo se encarga de controlar la traslación. El esquema obtenido de [1] descrito se puede observar en la figura 3.

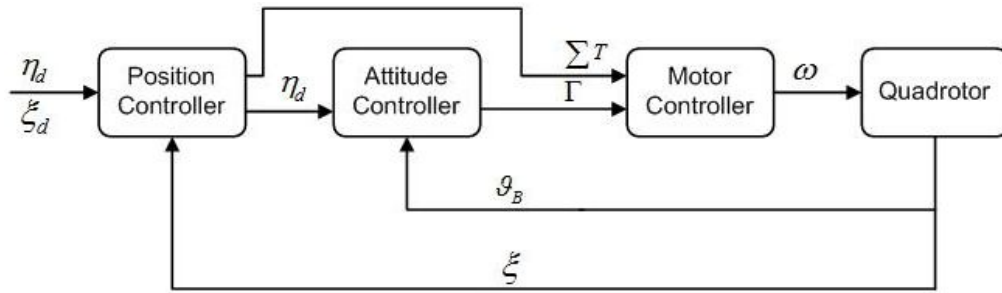


Figura 3: Esquema de control para drones quadrotores.

2.1.1. Controladores lineales

Los controladores lineales como bien se ha explicado anteriormente y se puede ver en la figura 2 engloba los controladores PID, LQR, Planificación de ganancias y H^∞ . En el apartado que se trata se hablará sobre las diferencias entre estos tipos de control y lo que aportan o no cada uno.

En cuanto a los PID se puede decir que estos controladores son los más eficientes y prácticos para el control en aplicaciones industriales. Su mayor ventaja es que pueden aplicarse sin la necesidad de conocer el modelo dinámico del dron y las ganancias PID se pueden calcular mediante prueba y error. Por eso este tipo de control es bastante extendido y muchos investigadores lo han desarrollado como se puede leer en [2], mientras que otros también han añadido sus propias variantes del método como [3].

LQR es un tipo de controladores óptimos y que consiguen un control muy robusto. Se desarrolló originalmente en el dron OS4 [2]. Primeramente para la implementación de este controlador no se tuvo en cuenta la dinámica lo que no daba resultados extremadamente buenos pero añadiendo la dinámica del dron este algoritmo se ha dotado de una gran robustez. Sin embargo se han podido encontrar en [3] y [4] variaciones de LQR como LQR híbrido, PID combinado con LQR, LQR con efecto integral o LQR con planificación de ganancias.

Los modelos no lineales de quadrotores normalmente se modelan como una combinación de modelos lineales cada uno con una región de operación diferente. Este modelado nos permite utilizar técnicas de control lineal para sistemas no lineales. Planificación de ganancias es una técnica de control basada en modelo que es ampliamente usada en diseño de control no lineal, también se ha podido combinar con otras técnicas de control para quadrotores [5].

H^∞ es un tipo de controlador muy robusto que es ampliamente usado en la práctica, se puede utilizar en sistemas que no tienen modelada la dinámica del sistema y con alguno parámetros inciertos. En la literatura se han aplicado a modelos linealizados de quadrotores. Un controlador lineal H^∞ basado en loop shaping se presentó en [6] y otros han presentado variantes para resolver el problema de seguimiento de caminos [7]

2.1.2. Controladores no lineales

En el apartado que se trata a continuación se comentará al igual que en el apartado anterior los controladores no lineales que se pueden observar en la figura 2.

La linearización de feedback es una técnica de control no lineal en la que las variables de estado del modelo dinámico son convertidas a un nuevo sistema de coordenadas a través de técnicas de transformación no lineales. El sistema resultante tiene una dinámica lineal por lo tanto técnicas lineales son aplicadas y después las coordenadas son convertidas mediante transformaciones inversas [8]. En [9] se presenta un esquema de control basado en linearización de feedback en la cual se diseñan un estimador y un observador de perturbaciones externas usando un orden mayor de sliding mode. Así también en [10] se desarrolla un sliding mode continuo basado en linearización de feedback.

Backstepping es una técnica renovada para sistemas de control subactuados. Este método ofrece un ratio de convergencia muy rápida y es capaz de soportar perturbaciones externas aunque carece de robustez como se puede comprobar en [11]. En [12] se ha propuesto este método haciendo uso de las ecuaciones de Lagrange para la dinámica.

SMC (Sliding Mode Control) es un método novel fácilmente aplicable a métodos de control no lineal que asegura robustez para perturbaciones conocidas. En [11] se aplica para el control de la orientación de un quadrotor con buenos resultados pero con unos resultados de vuelo por la media.

El control adaptativo es un método de control eficiente y robusto para sistemas que no tengan modelado la dinámica y que tengan incertidumbres sobre el modelo. Se presentó en [13] un controlador adaptativo no lineal para seguimiento de quadrotores mientras se tenía en cuenta rechazo de perturbaciones e incertidumbres.

MPC (Model Predictive Control) es otra técnica de control no lineal que también se ha utilizado para el control de MUAVs (Multirotor Unmanned Air Vehicle). MPC incorpora el modelo dinámico para prever los siguientes estados del sistema mientras que el error está siendo minimizado resolviendo los problemas de control óptimo. [14] propone un método para alternar el control MPC e implementa un controlador en un prototipo para realizar varios experimentos como tracking en interiores, flotar y maniobrar.

2.1.3. Controladores basados en aprendizaje

En este tipo de controladores el el modelo dinámico no es necesario ya que el sistema es entrenado usando datos obtenidos de experimentos de vuelo. Los controladores lógicos fuzzy y los controladores basados en redes neuronales se pueden encontrar en práctica actualmente.

Los controladores lógicos Fuzzy se han aplicado satisfactoriamente a los quadrotores en configuraciones standalone y combinándolos con otras técnicas de control. En [15] se presenta un controlador fuzzy-adaptativo para el problema de estabilización de un quadrotor

con carga y perturbaciones sinusoidales de viento. En cambio en [16] se propone un controlador fuzzy por cascada para un quadrotor en el que aunque se requiere ajustar parámetros se obtienen muy buenos resultados.

ANN (Artificial Neural Network) es también ampliamente utilizado en controladores inteligentes. En investigación se ha utilizado para identificación, estimación y control de sistemas. [17] utiliza ANN para aprender la dinámica de un sistema quadrotor y estimar los estados de este para proponer un feedback de control.

2.2. Sistemas de control y navegación experimentales

Otros métodos de navegación autónoma y control diferentes a los comentados anteriormente son los que se comentan a continuación. Se han buscado varios métodos pero se ha decidido explicar el funcionamiento de un algoritmo de navegación basado visión por computador [18] y otro basado en un sistema GNSS [19].

2.2.1. Usando visión por computador

Este método explica y presenta un algoritmo basado en visión por computador el cual está pensado y diseñado para funcionar en interiores. Este algoritmo se ha implementado en el dron AR.Drone el cual ofrece un sistema de navegación sofisticado y asistencia en varios tipos de maniobras como despegue, aterrizaje, mantenerse en una posición, etc... Para conseguir mantenerse en una posición en el aire así como para el control en posición este dron utiliza una cámara para estimar su velocidad horizontal.

Sin embargo para la tarea de estimar la velocidad horizontal utilizando la cámara se han usado dos algoritmos diferentes. Uno trata de hacer tracking de puntos de interés locales (utilizando un algoritmo llamado FAST corners) con diferentes frames y calcula la velocidad a partir del desplazamiento de estos puntos. Este algoritmo ofrece un resultado más preciso pero solo se puede utilizar cuando la velocidad del dron es baja y las texturas de la imagen están bien definidas.

El otro algoritmo que se utiliza estima la velocidad horizontal calculando el flujo óptico en imágenes piramidales. Este algoritmo es el utilizado durante el vuelo, es menos preciso pero es más robusto ya que no depende de las texturas del contraste de las escenas para funcionar. Además el AR.Drone usa información inercial obtenidas de las IMUs para estimar el estado del vehículo. De esta manera fusiona la información de la IMU con información de los algoritmos de visión y un modelo aerodinámico para estimar la velocidad del vehículo.

El control del AR.Drone es llevado a cabo de forma animada. El bucle interno controla la orientación del dron utilizando un controlador PID para calcular el ratio angular deseado basado en la diferencia entre la orientación actual y la orientación objetivo definida por el usuario. En el bucle exterior se utiliza un controlador proporcional para controlar los motores.

Cuando se liberan los controles, el AR.Drone calcula una trayectoria que lo llevará hasta una velocidad cero y un ángulo cero en un corto tiempo. Esta técnica se diseña off-line y usa un control feedforward la dinámica inversa del quadrotor.

2.2.2. Sistemas basados en GNSS

El algoritmo de navegación que se comenta a continuación hace uso de unas características que se explicarán a continuación. La primera característica es que utiliza un sistema de coordenadas llamado Geodetic. La posición en este sistema de referencia se define con 3 números (latitud, longitud y altitud) de la misma manera que se definen posiciones en globo terrestre como se puede ver en la figura 4.

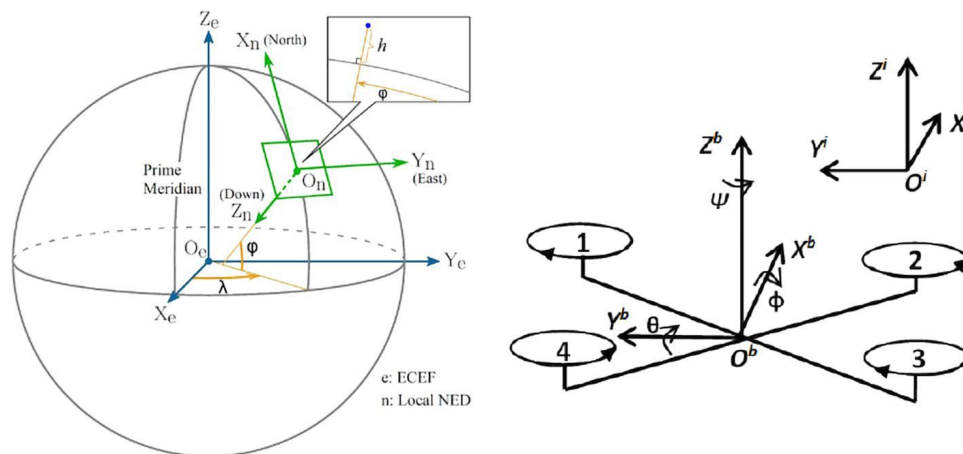


Figura 4: Sistema de coordenadas Geodetic (izquierda) y sistema de coordenadas del cuerpo (derecha)

El sistema de coordenadas para el cuerpo del dron tal y como se ve en la figura 4 muestra que el centro de coordenadas se sitúa sobre el centro de masas del dron. Haciendo uso de un sistema NED (North, East, Down) la X apunta hacia delante, la Y apunta hacia la derecha y la Z hacia abajo. Además también se definen dos conceptos uno es el “Ladeo” el cual se define como la diferencia angular entre dos coordenadas geodetic. Y el segundo concepto es la distancia Hubeny la cual calcula la distancia entre dos coordenadas Geodetic en el modelo terrestre.

El algoritmo de navegación que se comenta a continuación requiere una gran cantidad de datos de satélites por esta razón es por la que se usa GNSS y una brújula que pueden conectar más variaciones de los satélites. El sistema es un sistema para transporte de paquetes por eso también cuenta con un algoritmo de entrega de paquetes que se basa en un sensor que es capaz de detectar cuando se carga un paquete y así saber que cuando aterrice debe descargar el paquete antes de comenzar el vuelo de vuelta.

El algoritmo de navegación en sí, es un método que calcula y planea una trayectoria desde una posición actual hasta una posición objetivo usando el “Ladeo” explicado anteriormente. El valor de ladeo se compara continuamente con frente del dron ya que el sistema propuesto

aplica la velocidad según el sistema de coordenadas. Cuando el dron está volando hacia la posición cada pocos segundos se recalcula la trayectoria con la posición actual. Además el sistema comprobará continuamente la distancia entre el punto objetivo y el punto actual (distancia de Hubeny). Si la distancia calculada es menor que 5 m se reducirá la velocidad y si es menor que 0.5 m entonces se aterrizará. Y este proceso iterativo se corre cada vez que llega información de la posición del dron.

3. Sistema de detección

El primer procedimiento que se verá es el sistema de detección que se utiliza para poder obtener la información referida a la localización del dron en el mundo tridimensional en el que se encuentra. Este sistema se basa en dos elementos principales, que se procederán a explicar en los apartados siguientes, que son los códigos QR utilizados como “*landmarks*” y el sistema de visión formado por cámaras estereoscópicas.

3.1. Códigos QR

Como se ha mencionado anteriormente, el sistema de localización se basa en la detección de diversos códigos QR repartidos por el entorno donde operará el dron para que de esta forma el dron pueda estimar su posición relativa. Gracias a estos elementos se puede codificar información como la posición que ocupan en referencia a un sistema de referencia global, su orientación o el tamaño total del mismo. Por consiguiente, al enfocar uno de estos elementos con el sistema de visión del dron se podrá decodificar toda los datos mencionados.

Como se explica en [20], en el entorno experimental hay que tener en cuenta la orientación de este elemento, ya que se pueden producir errores de correspondencia. Es por ello que la estructura del código QR nos indica la orientación correcta de la siguiente forma:



Figura 5: Ejemplo de código QR y descripción de su orientación

Por otro lado, este proceso que permite codificar y obtener la información de los códigos QR se gestiona a través de librerías como pueden ser “*zbar*” o “*ZXing*”, ambas de dominio público. En concreto, la primera de ellas nos permite detectar los códigos QR en un frame cualquiera y devolver la información codificada que contienen y las coordenadas de sus esquinas.

En lo referente a su funcionamiento, el primer paso del procedimiento consiste en convertir la imagen obtenida por la cámara a blanco y negro, para que de esta forma su análisis sea más sencillo. Una vez obtenida la imagen deseada y pasándola a un tipo de dato que maneja la biblioteca, conocido como “*imagenZBar*”, ya se podrá escanear la imagen en busca del código QR. Cuando se localiza, se guarda las coordenadas de las esquinas en la imagen y la etiqueta que guarda, la cual podrá ser analizada para obtener los valores deseados. De esta forma se pueden almacenar tanto los valores de las esquinas en la imagen como los valores de las esquinas en el entorno 3D.

Así mismo, la información anterior nos permite establecer elementos visuales en la representación de la cámara del robot, de modo que se pueda verificar el funcionamiento del mismo al mismo tiempo que se proporciona al observador información adicional. En la figura siguiente se puede ver un ejemplo de este tipo de procesos, donde se delimita el código QR.



Figura 6: Detección de un código QR

3.1.1. Aplicaciones prácticas

Con la información anterior como base, es interesante examinar algunas de las aplicaciones prácticas desarrolladas para robots móviles, de forma que podamos ver la utilidad real de este sistema. Una de las maneras que se han utilizado los códigos QR para guiar a robots se ve en el [21], donde se sitúan en el techo del entorno y con la ayuda de una cámara posicionada hacia arriba. Al ser una sola cámara, la posición en el eje Z no se puede calcular, pero al ser robots terrestres se asume que es constante. Por lo tanto, tras un paso previo de calibración de la cámara, el robot será capaz de decodificar los datos de los distintos códigos QR que indican la posición central de QR y su orientación. De esta forma puede calcular su posición relativa respecto a esta información de posicionamiento. En conclusión, se obtiene un método de localización del robot gracias a una malla de códigos QR situada en el techo, como se puede ver en la imagen.

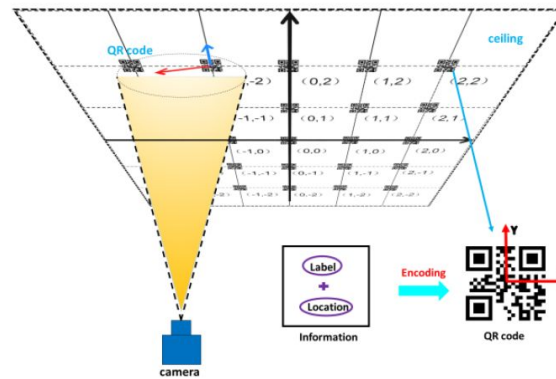


Figura 7: Representación de la lectura de códigos QR

Otra de las maneras que se utilizan esta tecnología es como soporte para otros sistemas de localización como puede ser el uso de la odometría, visto en [22]. En este caso se utilizan dos encoders para obtener datos de la odometría, mientras que con una cámara monocular se detectan códigos QR situados en el suelo, como se puede ver a continuación:

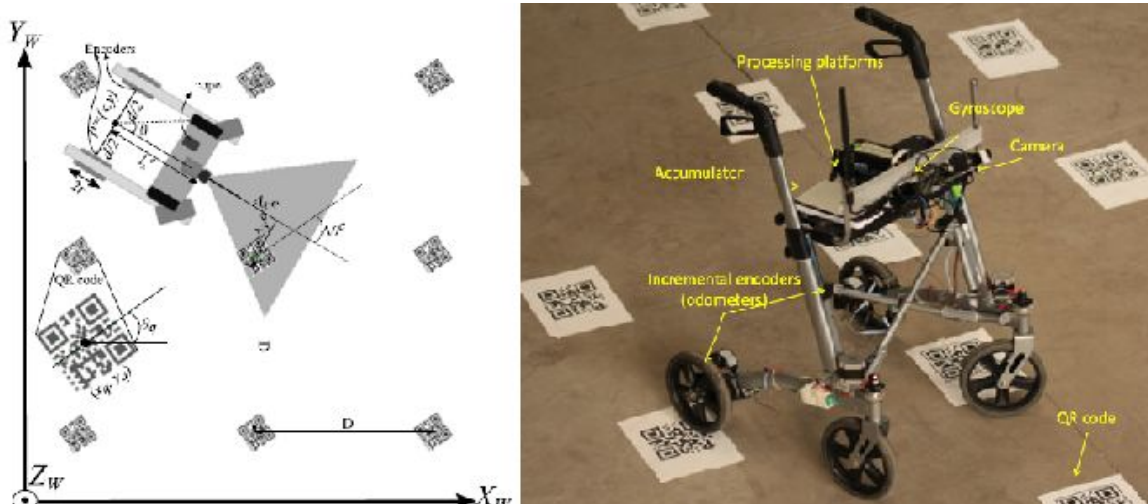


Figura 8: Sistema de localización con Odometría y códigos QR

El papel del sistema de visión y los códigos QR es reducir el error acumulado en los cálculos de la odometría, ya que al tratarse en términos generales de un mapa de “*landmarks*”, se puede utilizar la información de posición de los códigos QR para obtener la trayectoria correcta seguida por el robot. Además, de su posición en el espacio, los códigos QR también contienen el dato de su orientación respecto a un sistema global, lo que ayudará a conocer la orientación del robot.

En ambos casos no se tiene en cuenta la profundidad a la que se encuentra el código QR debido a los propios sistemas utilizados que solo se mueven en dos dimensiones. Sin embargo, en casos de utilizar movimientos tridimensionales es necesario aplicar un sistema de visión diferente a las cámaras monoculares, es decir, la visión estereoscópica. Este método se explicará en el apartado siguiente, así como, se presentarán aplicaciones en las que se utilice.

3.2. Visión estereoscópica

La obtención de la distancia ha sido una preocupación constante en algunos sistemas, particularmente en los diseñados para la navegación en robótica, donde es necesario conocer en cada momento la estructura de la escena para navegar en el entorno. Con tal propósito se han desarrollado ciertas técnicas y dispositivos, como puede ser la visión estereoscópica artificial, donde dos o más cámaras captan la misma escena y por triangulación se determina la distancia.

Su funcionamiento se basa en la utilización de dos o más cámaras, separadas entre sí a una cierta distancia constante, ya que la modificación de esta distancia durante el proceso provocará un error que solo se puede solucionar volviendo a hacer el proceso de calibrado previo. Cada imagen capturada desde cada una de las posiciones de las cámaras presentarán un cierto desplazamiento de la escena real, lo que permitirá obtener la distancia a la que se encuentra determinado objeto del mundo. Una disposición posible para un robot móvil se muestra a continuación:

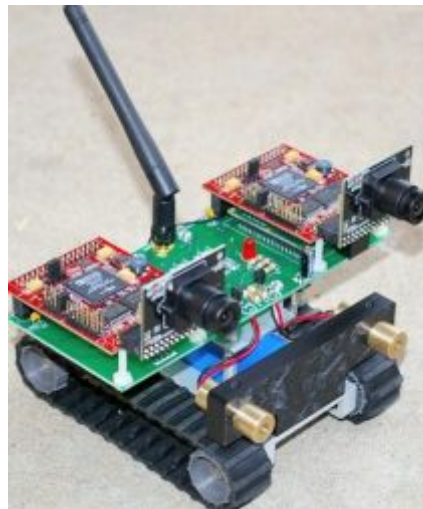


Figura 9: Disposición de un sistema de visión estereoscópica

Una implementación práctica de este concepto que podemos encontrar en [23] es el proyecto de un robot móvil encargado de realizar acciones simples en una librería, como la búsqueda de libros y su colocación. Estas acciones llamadas operaciones BAR (*Book accessing and returning*) serán llevadas a cabo por un robot formado por dos partes. La primera será la plataforma de movimiento que permitirá al robot dirigirse a las zonas designadas. Mientras, el cuerpo del robot montado encima de la plataforma contará con diversos dispositivos para realizar estas tareas, como el sistema de navegación inercial para gestionar sus movimientos, el sistema de visión estereoscópico que estará situado sobre los brazos del robot. En la siguiente imagen se puede observar la estructura comentada:

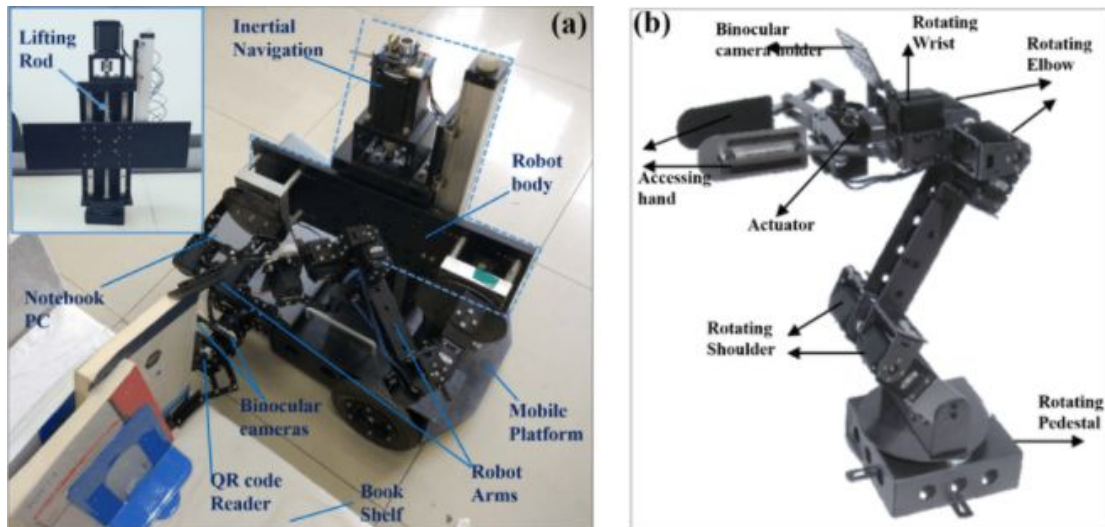


Figura 11: Estructura del robot

A la hora de llevar a cabo la manipulación de los libros, primeramente el robot realizará un acercamiento del brazo robot a una posición cercana al libro de forma que la cámara pueda detectar el código QR del libro. Una vez escaneado, el robot podrá obtener información detallada del libro, así como, la posición y orientación del libro en el espacio 3D. Con estos datos el robot finalmente podrá calcular la trayectoria idónea para manipular el libro, incluso hacer funcionar los dos brazos cooperativamente en la realización de la tarea.

4. Sistema de localización

La ventaja de utilizar un sistema de visión estereoscópica binocular para la localización del robot frente a un sistema monocular, es que permite obtener las coordenadas 3D de un punto que se encuentre en las dos imágenes. Utilizando un sistema de una única lente sólo se pueden obtener dos coordenadas, porque no es posible computar la profundidad.

Es posible implementar un algoritmo de localización de un robot móvil utilizando un sistema monocular, pero restringe la posición de los QR porque tienen que estar colocados en un plano paralelo al plano de la imagen, para mantener constante la coordenada de profundidad. Sin embargo, aumentando el número de cámaras la posición de los QR no está restringida, ya que el sistema es capaz de calcular las coordenadas 3D de los puntos.

La mayoría de desarrollos en los que se utilizan sistemas de visión binocular para robots móviles están enfocados en extracción de características. Por ejemplo, en [24] se utiliza este sistema para clasificar objetos, que puede ser una subtask de un robot móvil con ciertos protocolos de manipulación de objetos. En otros trabajos, como por ejemplo en [25] se utiliza este sistema visual en conjunto con técnicas de control y CNNs para poder simular el comportamiento humano para localizar y dirigirse a un objetivo, focalizando la atención en él utilizando segmentación visual. En los trabajos enfocados a tareas propias de la robótica móvil, estos sistemas se utilizan para extraer características, como por ejemplo en [26], que utilizan este sistema para extraer características del entorno y poder realizar el proceso de SLAM.

Las ventajas de utilizar este sistema en un paradigma de localización junto a los QR es que es posible implementar un método robusto, versátil, flexible, económico y computacionalmente muy eficiente. Al igual que para cualquier aplicación que utilice sistemas de visión, el primer paso siempre es la calibración de las cámaras y la obtención de sus parámetros intrínsecos.

4.1. Calibración

Un sistema de visión estereoscópica binocular está formado por dos cámaras separadas una distancia d en el eje X normalmente. Cada una de ellas se puede modelar independientemente como un sistema de modelo *pinhole*. El esquema de dicho modelo es el siguiente:

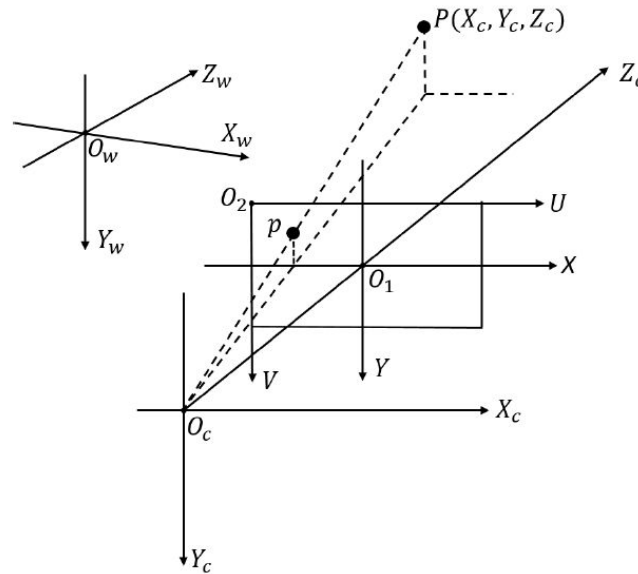


Figura 12: Modelo *pinhole* para la calibración de las cámaras

Donde O_w es el sistema de coordenadas del mundo; O_c , el de la cámara; y $O_1 - XY$ y $O_2 - UV$, el sistema de coordenadas del plano imagen y el sistema de coordenadas del plano en píxeles, respectivamente. Podemos encontrar el desarrollo completo del proceso de calibración en [23], pero la ecuación resultante para obtener las coordenadas de un punto del mundo 3D en el espacio de la imagen es:

$$\begin{aligned}
 Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\
 &= M_1 M_2 X_W = M X_W,
 \end{aligned}$$

En la primera igualdad se pueden definir las matrices. La primera matriz corresponde al cambio entre el sistema O_1 y O_2 y la segunda matriz corresponde al cambio entre el sistema O_c y O_1 . Estas dos matrices se multiplican para obtener la matriz de **parámetros intrínsecos**

de la cámara. Esta matriz representa el cambio entre las coordenadas del punto P en el sistema de coordenadas 3D de la cámara y el sistema de coordenadas 2D del plano imagen en píxeles. Los parámetros de esta matriz son únicos para cada cámara y se obtienen durante el proceso de calibración de la misma. Una vez obtenidos, se mantienen fijos mientras se mantienen las características de la cámara. La tercera matriz de la primera igualdad es la matriz de transformación homogénea entre el sistema de coordenadas del mundo y el sistema de coordenadas de la cámara, es decir, describe la posición de la cámara en el mundo. Esta matriz recibe el nombre de **parámetros extrínsecos**, y cambia cada vez que se modifica la posición de la cámara respecto del mundo, que en robots móviles va a ser frecuente.

El modelo pinhole no tiene en cuenta la deformación producida por la curvatura de las lentes de la cámara, por lo que hay que corregir la distorsión angular producida utilizando las siguientes ecuaciones:

$$\begin{cases} \bar{x} = x + \delta_x = x(1 + k_1 r^2) \\ \bar{y} = y + \delta_y = y(1 + k_2 r^2) \end{cases}$$

Para la calibración de las cámaras, el método más utilizado es **OpenCV**, disponible para python y C++, que es una gran librería que ofrece métodos para muchas otras subtarefas de visión por computador, entre ellas la calibración de las cámaras. También existen otras librerías como *toolbox* de Matlab, aunque son menos comunes de encontrar.

4.2. Triangulación

Una vez obtenidos los parámetros intrínsecos de la cámara mediante el proceso de calibración, se puede establecer la relación entre las coordenadas de un punto P en la imagen y sus coordenadas en el sistema de referencia de la cámara, utilizando la triangulación. Representando el sistema de visión como:

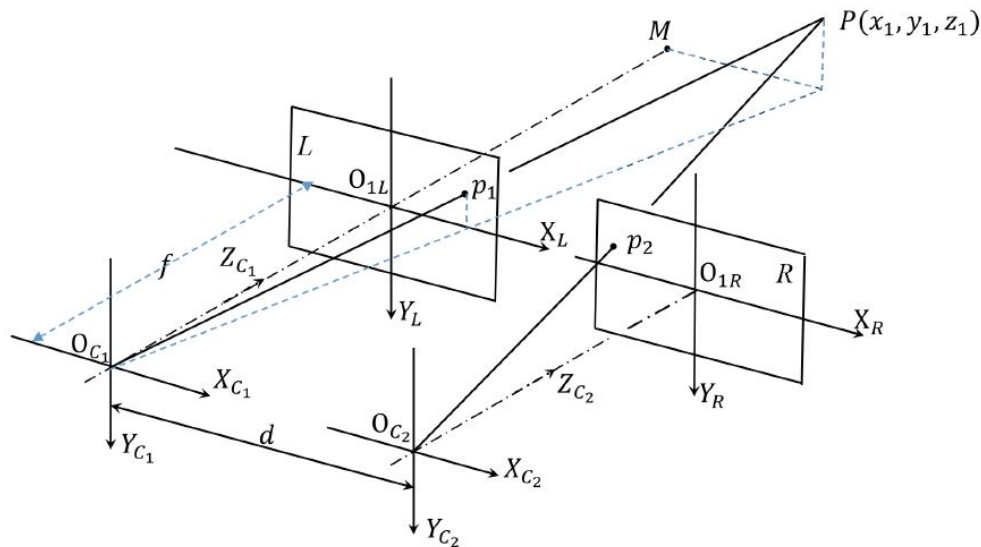


Figura 13: Esquema del sistema de visión binocular estereoscópico

Se pueden definir las coordenadas 3D en el sistema de referencia de la cámara del punto P como:

$$\begin{cases} X = x_1 = \frac{d(u_1 - u_0)}{u_1 - u_2} \\ Y = y_1 = \frac{df_x(v_1 - v_0)}{f_y(u_1 - u_2)} \\ Z = z_1 = \frac{df_x}{u_1 - u_2} \end{cases}$$

Las dos únicas variables que no se obtienen directamente del proceso de calibración son u_1 y u_2 . Estas variables representan la coordenada u del punto P en los dos sistemas de coordenadas del plano imagen en píxeles. Por lo tanto, para que el sistema funcione es necesario emparejar el punto P en la imagen de la cámara izquierda con su análogo en la imagen de la cámara derecha. Este problema de *matching* es una de las principales limitaciones del uso de sistemas binoculares, ya que requiere extracción de características, emparejamiento y algoritmos de reducción de errores de emparejamiento. Estos métodos añaden complejidad computacional y temporal al sistema, disminuyendo su eficiencia. Al utilizar la localización con códigos QR este problema desaparece.

4.2.1. Matching

En esta aplicación se utilizan códigos QR junto al sistema de visión binocular para obtener las coordenadas de un punto P en el sistema de coordenadas de la cámara. Para ello, cuando se detecta un QR se obtienen las coordenadas en el espacio de la imagen de cada una de las esquinas. En esta aplicación no es necesario un proceso de *matching* ni de extracción de características, porque directamente con la librería de detección de los códigos tenemos la información de cada uno de los puntos en cada imagen. Como estos puntos se devuelven en forma de vector y están ordenados siempre de la misma forma, se hace el emparejamiento entre esquinas directamente. De esta forma, aplicando las ecuaciones del apartado anterior, podemos obtener las coordenadas 3D de cada una de las esquinas del QR respecto al sistema de referencia de la cámara.

Para evitar errores, es aconsejable añadir una variable más a la información codificada en el QR, que corresponde con una identificación de dicho QR. De esta forma, si se está visualizando más de un código en una imagen, se comparan las esquinas de los QR que tienen el mismo identificador.

4.3. Transformaciones

Una vez obtenidas las coordenadas de las esquinas del QR respecto al sistema de referencia de la cámara, utilizando la información codificada en el propio QR, que son las coordenadas de cada una de las esquinas respecto al sistema de referencia global, es posible obtener las coordenadas de la cámara respecto al sistema de referencia global. El sistema funciona con la información de una sola esquina del QR, pero si tenemos las cuatro podemos hacer un promedio de la posición de la cámara obtenida y así disminuir posibles errores.

Podemos definir las ecuaciones para obtener las coordenadas (x_c , y_c , z_c) de la posición de la cámara respecto al sistema de referencia del mundo como:

$$x_c = p_{x_{qr}} - p_{x_c}$$

$$y_c = p_{y_{qr}} - p_{y_c}$$

$$z_c = p_{z_{qr}} - p_{z_c}$$

Siendo p_{qr} las coordenadas del punto P obtenidas de la decodificación del QR y p_c las coordenadas del punto P obtenidas del sistema de visión binocular; y suponiendo que los ejes de ambos sistemas de referencia están alineados (no hay rotaciones entre ellos). Si no estuvieran alineados, las relaciones anteriores se modificarían en función de dicha alineación.

Tras todo el proceso, se obtienen las coordenadas de la cámara respecto al sistema de referencia del mundo. En función de la geometría del robot móvil puede ser interesante obtener las coordenadas de otro punto más significativo, cómo por ejemplo el centro geométrico o el centro de masas. En este caso, sería necesario realizar otra transformación entre el sistema de referencia de la cámara y el sistema de referencia del robot, que será determinada por las medidas reales del robot en el que se vaya a implementar el sistema y la posición de las cámaras en dicho robot. En el siguiente esquema se muestra los distintos sistemas de referencia que se pueden encontrar una vez obtenidas las coordenadas del punto P en el sistema de referencia de la cámara:

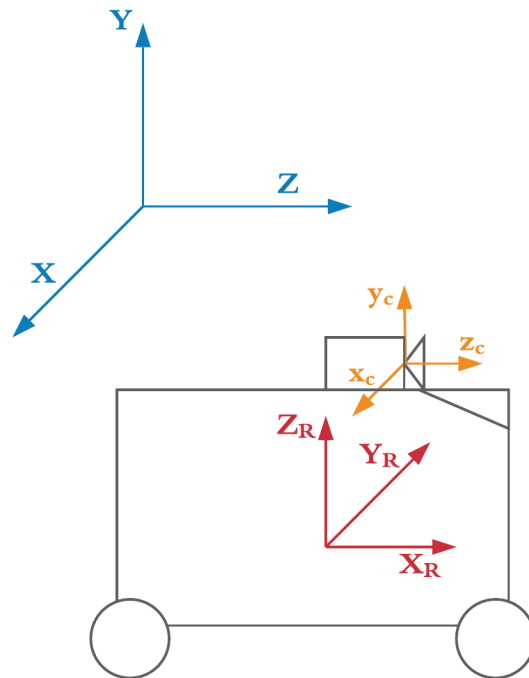


Figura 14: Esquema de los sistemas de referencia de la tarea una vez determinadas las coordenadas de P .

5. Referencias

- [1] Amin, Roohul; Aijun, Li; Shamshirband, Shahaboddin (2016). A review of quadrotor UAV: control methodologies and performance evaluation. *International Journal of Automation and Control*, 10(2), 87–. doi:10.1504/ijaac.2016.076453
- [2] Bouabdallah, S., Noth, A. and Siegwan, R. (2004) ‘PID vs. LQ control techniques applied to an indoor micro quadrotor’, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2451–2456
- [3] Argentim, L.M., Rezende, W.C., Santos, P.E. and Aguiar, R.A. (2013) ‘PID, LQR and LQR-PID on a quadcopter platform’, 2013 *International Conference on Informatics, Electronics and Vision, ICIEV 2013* [online] <http://doi.org/10.1109/ICIEV.2013.6572698> (accessed 28 June 2015).
- [4] Peña, V.G.M., Vivas, C.G.E. and Rodriguez, I.F.C. (2012) ‘Simulation of the quadrotor controlled with LQR with integral effect’, *ABCM Symposium Series in Mechatronics*, Vol. 5, No. 1, pp.390–399.
- [5] Milhim, A.B., Zhang, Y., Rabbath, C., Student, M. and Scientist, D. (2010) ‘Gain scheduling based PID controller for fault tolerant control of a quad-rotor UAV’, *AIAA Infotech@Aerospace*, April, pp.1–13.
- [6] Chen, M. and Huzmezan, M. (2003) ‘A combined MBPC/2 DOF H_∞ controller for a quad rotor UAV’, *AIAA Atmospheric Flight Mechanics Conference and Exhibition*, August [online] http://pdf.aiaa.org/preview/CDReadyMGNC03_664/PV2003_5520.pdf (accessed 15 June 2015).
- [7] Mokhtari, A., Benallegue, A. and Daachi, B. (2005) ‘Robust feedback linearization and H_∞ controller for a quadrotor unmanned aerial vehicle’, 2005 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, (IROS 2005), pp.1198–1203 [online] <http://doi.org/10.1109/IROS.2005.1545112> (accessed 26 June 2015).
- [8] Mistler, V., Benallegue, A. and Sirdi, N.K.M. (2001) ‘Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback’, *IEEE International Workshop on Robot and Human Interactive Communication*.
- [9] Benallegue, A., Mokhtari, A. and Fridman, L. (2006) ‘Feedback linearization and high order sliding mode observer for a quadrotor UAV’, *Proceedings of the International Workshop on Variable Structure Systems*, pp.365–372.
- [10] Fang, Z., Zhi, Z., Jun, L. and Jian, W. (2008) ‘Feedback linearization and continuous sliding mode control for a quadrotor UAV’, *Proceedings of the 27th Chinese Control Conference*, pp.349–353

- [11] Bouabdallah, S. and Siegwart, R. (2005) ‘Backstepping and sliding-mode techniques applied to an indoor micro quadrotor’, *Proceedings – IEEE International Conference on Robotics and Automation*, April, pp.2247–2252 [online] <http://doi.org/10.1109/ROBOT.2005.1570447> (accessed 5 June 2015).
- [12] Das, A., Lewis, F. and Subbarao, K. (2009) ‘Backstepping approach for controlling a quadrotor using Lagrange form dynamics’, *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 56, Nos. 1–2, pp.127–151 [online] <http://doi.org/10.1007/s10846-009-9331-0> (accessed 1 June 2015).
- [13] Islam, S., Liu, P.X. and El Saddik, A. (2014) ‘Nonlinear adaptive control for quadrotor flying vehicle’, *Nonlinear Dynamics*, Vol. 78, No. 1, pp.117–133 [online] <http://doi.org/10.1007/s11071-014-1425-y> (accessed 26 June 2015).
- [14] Alexis, K., Nikolakopoulos, G. and Tzes, A. (2012) ‘Model predictive quadrotor control: attitude, altitude and position experimental studies’, *IET Control Theory & Applications*, Vol. 6, No. 12, pp.1812–1827 [online] <http://doi.org/10.1049/iet-cta.2011.0348> (accessed 28 June 2015).
- [15] Coza, C. and Macnab, C.J.B. (2006) ‘A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization’, *Annual Conference of the North American Fuzzy Information Processing Society – NAFIPS*, pp.475–479 [online] <http://doi.org/10.1109/NAFIPS.2006.365452> (accessed 28 May 2015).
- [16] Szlachetko, B. and Lower, M. (2012) ‘On quadrotor navigation using fuzzy logic regulators’, in *Computational Collective Intelligence. Technologies and Applications*, pp.210–219, Springer, Berlin Heidelberg
- [17] Dierks, T. and Jagannathan, S. (2010) ‘Output feedback control of a quadrotor UAV using neural networks’, *IEEE Transactions on Neural Networks*, A Publication of the IEEE Neural Networks Council, Vol. 21, No. 1, pp.50–66 [online] <http://doi.org/10.1109/TNN.2009.2034145> (accessed 5 June 2015).
- [18] Jiménez Lugo, J., Zell, A. Framework for Autonomous On-board Navigation with the AR.Drone. *J Intell Robot Syst* 73, 401–412 (2014). <https://doi.org/10.1007/s10846-013-9969-5>
- [19] Patrik, A., Utama, G., Gunawan, A.A.S. et al. GNSS-based navigation systems of autonomous drone for delivering items. *J Big Data* 6, 53 (2019). <https://doi.org/10.1186/s40537-019-0214-3>

- [20] R. Taketani and H. Kobayashi, "A Proposal for Improving Estimation Accuracy of Localization Using QR codes and Image Sensors," IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 2019, pp. 6815-6820, doi: 10.1109/IECON.2019.8927589
- [21] H. Zhang, C. Zhang, W. Yang and C. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 2501-2506, doi: 10.1109/ROBIO.2015.7419715.
- [22] P. Nazemzadeh, D. Fontanelli, D. Macii and L. Palopoli, "Indoor Localization of Mobile Robots Through QR Code Detection and Dead Reckoning Data Fusion," in IEEE/ASME Transactions on Mechatronics, vol. 22, no. 6, pp. 2588-2599, Dec. 2017, doi: 10.1109/TMECH.2017.2762598.
- [23] Yu, X., Fan, Z., Wan, H., He, Y., Du, J., Li, N., Yuan, Z., & Xiao, G. (2019). Positioning, Navigation, and Book Accessing/Returning in an Autonomous Library Robot using Integrated Binocular Vision and QR Code Identification Systems. *Sensors* (Basel, Switzerland), 19(4), 783. doi: 10.3390/s19040783
- [24] G. Huang and C. Cheng, "3D coordinate identification of object using binocular vision system for mobile robot," 2013 CACS International Automatic Control Conference (CACS), Nantou, 2013, pp. 91-96, doi: 10.1109/CACS.2013.6734113.
- [25] Jian Peng, A. Srikaew, M. Wilkes, K. Kawamura and A. Peters, "An active vision system for mobile robots," Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0, Nashville, TN, 2000, pp. 1472-1477 vol.2, doi: 10.1109/ICSMC.2000.886062.
- [26] L. Gao, Y. Gai and S. Fu, "Simultaneous Localization and Mapping for Autonomous Mobile Robots Using Binocular Stereo Vision System," 2007 International Conference on Mechatronics and Automation, Harbin, 2007, pp. 326-330, doi: 10.1109/ICMA.2007.4303563.