Desarrollo inicial de un ambiente de competencia y experimentación en robótica situada con drones aplicado a la formación de estudiantes en el contexto de resolución de misiones.

Diego Avila[1], Emiliano Lorusso[1], Sofía Fasce[1], Jorge Ierache[1]

[1] Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica, Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales, Universidad de Morón, Morón, Argentina

jierache@unimoron.edu.ar, adiego73@gmail.com, sofiafasce@gmail.com , lorusso.emiliano@gmail.com

Recibido: 09/11/2016 | Revisado: 10/04/2017 | Aceptado: 21/04/2017

Resumen

El presente trabajo introduce las características de las competencias de robótica situada aplicadas al fútbol de robots como base inicial de la temática, para luego presentar el desarrollo de un ambiente de competencia y experimentación en robótica situada orientado a la utilización de un drone, en particular cuadricóptero en el contexto de la formación de estudiantes de nivel secundario y universitario.

Palabras clave: Drone; Robótica situada; Competición; formación.

Abstract

The current paper introduces the situated robotics competitions characteristics applied to robot soccer as the basis of the current subject, and exposes the development of a competition and experiment environment of situated robotics with the aiming on the use of a drone, in the context of high school and university student's development.

Keywords: Drone, Situated robotics; Competition; student development.

1. Introducción

1.1. Robótica Situada

El control de robots autónomos ha sido de gran interés desde los comienzos de la robótica y la inteligencia artificial. Para poder experimentar el control autónomo de robots, se han desarrollado diversos ambientes de robótica situada orientados a la competición, un ejemplo de esto es el fútbol de robots [1] y [2], o las ligas de sumo [3]. La robótica situada se caracteriza por: a) atender las problemáticas de captación del ambiente de actuación del robot en tiempo real (sistema de visión artificial y sensores asociados); b) generar una navegación precisa en un contexto de actuación del robot autónomo; c) desarrollar las acciones del robot en tiempo real en un ambiente dinámico; d) actuar en forma colaborativa, en particular en aplicaciones como el fútbol de robots, enfrentando estrategias en oposición. Con la finalidad de introducir la temática de competencias de robótica situada, se desarrolla a continuación una descripción general de las competencias y categorías de Fútbol de Robots.

El fútbol de robots reúne a diversas organizaciones y grupos de investigación han creado ámbitos de competencia relacionados con la robótica situada. Una de estas organizaciones es FIRA [2] (Federation of International Robot-soccer Association), que durante años ha organizado distintas competencias con eje en el fútbol de robots, convocando a estudiantes de nivel

secundario y universitario de todo el mundo, destacándose las categorías de competencia MiroSot [4] y NaroSot [5]. Estas competencias ayudan a que los participantes innoven en determinados campos como: comportamiento emergente, visión artificial, inteligencia artificial, etc.

MiroSot y NaroSot son las categorías de fútbol de robots utilizando cinco u once robots con dos ruedas. Cada equipo debe contar con una computadora que procese el video proveniente de una cámara suspendida e identifique la ubicación de los robots y la pelota. Dentro de la categoría MiroSot, existen dos sub-categorías: la MiroSot Middle League, donde se enfrentan dos equipos de cinco robots, y la MiroSot Large League, donde se enfrentan dos equipos de once robots. En la figura 1 se puede observar un robot de la categoría MiroSot

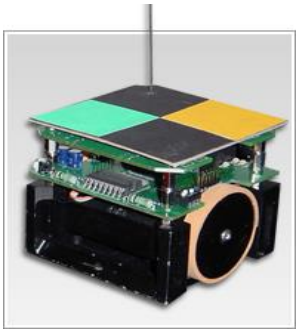

Figura 1. Robot de la categoría MiroSot.

RoboSot [6] es una categoría de fútbol de robots que consiste en dos equipos de entre uno y tres robots completamente autónomos en su sistema de visión o semi-autónomos. Los autónomos deben ser capaces de procesar y analizar la información proveniente de su sistema de visión. Los robots semi-autónomos, en cambio, el procesamiento del video se realiza en una computadora externa. En esta categoría, los robots de cada equipo deben tener como máximo un tamaño de 20 cm x 20 cm, sin límite de altura.

Otra de las ligas de fútbol de robots es la F180 [7] que se centra en el problema de inteligencia, cooperación y control multi-agente en un entorno de robótica situada. También denominada Small Size League (SSL), participan dos equipos con seis robots cada uno y cada uno de estos debe cumplir con las especificaciones de tamaño, es decir que debe caber dentro de un círculo de 180mm de diámetro y no puede superar 15cm de altura. Los robots juegan de forma autónoma en una cancha de 6.05m de largo y 4.05m de ancho cubierta por alfombra color verde y con una pelota de golf color naranja. En la figura 2 se observa un partido de esta categoría, donde se pueden ver tanto los robots con sus parches de color para poder identificarlos. Todos los objetos en el campo son seguidos por un sistema de visión estandarizada, llamado SSL Visión, que procesa los datos obtenidos por dos cámaras que se encuentran instaladas a 4m de altura del campo de juego

Figura 2. Robots de la categoría F180 en el ambiente de competencia.

En el transcurso del encuentro los robots utilizan comunicación inalámbrica mediante la cual la computadora central, que está fuera del campo, les envía información sobre su posición, a la estrategia del juego. Tanto en la categoría RoboSot como en la F180 el desarrollo involucra investigación en el campo de la visión artificial, y particularmente, en áreas relacionadas a la identificación, localización y/o clasificación de elementos en el ambiente; en el campo de los ambientes cooperativos, donde los robots exhiben capacidades para interactuar entre sí; y en el campo de la navegación, donde los robots exhiben capacidades en el traslado de la pelota o planeamiento de rutas.

1.2. Ambientes de Competencia con Drones

En el contexto de multirotores, la International Aerial Robotics Competition (IARC) [8], es una competición anual organizada por el Instituto Tecnológico de Georgia desde 1991. Esta competencia está orientada a vehículos aéreos de cualquier tipo (helicópteros, cuadricópteros, etc.) y tiene por objetivo principal proporcionar una razón para que el estado del arte en la materia avance. Desde 1991 hasta 2009, se propusieron un total de 6 misiones que involucran robots completamente autónomos, hasta ese momento inimaginables. En 2013, se propuso la séptima misión que involucra robots completamente autónomos, pero con el agregado de que deben interactuar con otros robots en tierra o incluso competir a la vez con otros robots aéreos. Esta competencia involucra todo tipo de aspectos relacionados a los robots aéreos, situados en un ambiente especial, deben ser capaces de reconocerse mutuamente o interactuar con robots terrestres o aéreos, para completar la misión asignada (figura 3).

Figura 3. Robot de la Universidad Técnica de Berlin realizando la 3er misión de la competencia IARC

Otra competencia similar es la Student Unmanned Aerial Systems Competition (SUAS) [9]. Esta competencia tiene por objetivo incentivar a los alumnos de carreras técnicas a interesarse en los sistemas aereos no tripulados. En estas competencias, los alumnos deberán diseñar un sistema de vuelo no tripulado, y demostrar la capacidad del mismo de volar de forma desatendida para realizar una misión dada. La competencia se realiza al aire libre, y permite cualquier tipo de robot siempre que sea completamente autónomo y aéreo.

En la Argentina, si bien existen algunas competencias relacionadas con la robótica situada, no existen competencias de robots aéreos autónomos no tripulados. Creemos que es importante que haya este tipo de competencias en el país, con el objetivo de incentivar a los alumnos a interesarse en la robótica y particularmente, en los robots aéreos autónomos.

2. Desarrollo del ambiente de Robótica Situada Aplicada a Drones

El uso de drones para realizar diferentes tipos de tareas es de especial interés en la robótica autónoma. Se requiere una cantidad de parámetros que se deben tener en cuenta a la hora de moverse por el ambiente, así como también sus grados de libertad. Se plantea como problema la generación de un ambiente de robótica situada utilizando drones, considerando el control autónomo de los mismos a partir de información del ambiente provista por la cámara suspendida y la utilización de diferentes sensores disponibles en el drone. El modelo del ambiente de navegación utilizado para el desarrollo de pruebas y competencias de vuelo se representa en la figura 4, donde se observa un campo de vuelo de 3 metros por 2 metros, definido por el ángulo de visión de la cámara que está colocada en el centro del campo a 4 metros de altura. Dentro del campo, se sitúan cuatro objetivos a alcanzar por el drone, representados e identificados por parches de colores diferentes e irrepetibles circulares, que son identificados a través del análisis de la imagen capturada por la cámara suspendida. A modo de ejemplo consideramos: el checkpoint “A”, de color amarillo; el checkpoint “B” de color verde; el checkpoint “C” de color naranja; y el checkpoint “D”.

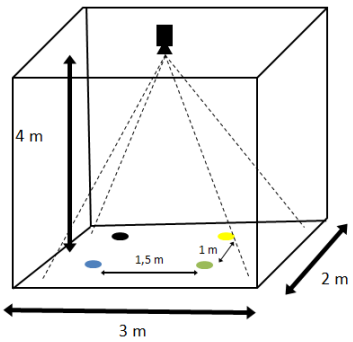

Figura 4. Ambiente utilizado

Tanto la información del ambiente como la del robot se le exhiben al usuario por medio de la interfaz gráfica de visualización. En la figura 5, se puede apreciar la misma con los datos que presenta: la posición de los checkpoints (a), la posición del robot (b), el área visible por la cámara (c), los valores relativos a los sensores del robot (d) y unos gráficos que muestran los distintos valores resultantes del PID (e).

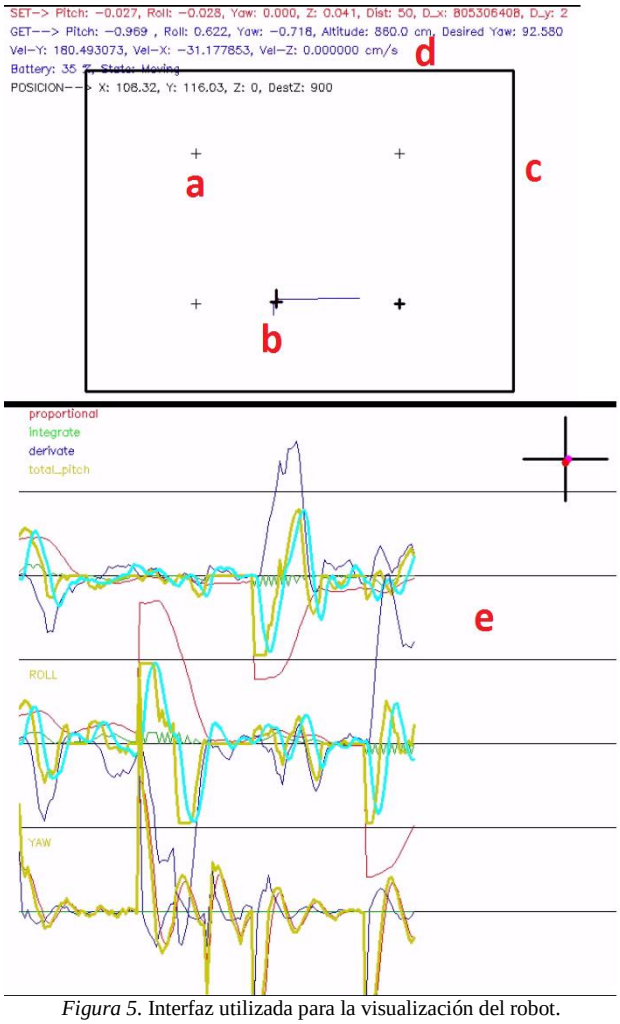

Figura 5. Interfaz utilizada para la visualización del robot.

El control del drone se realiza mediante el análisis del video proveniente de la cámara suspendida, corrigiendo la trayectoria del drone para mantenerlo estable. Se utilizó inicialmente un robot desarrollado por un grupo originario de Suecia, llamado CrazyFlie [10], que facilitó al equipo de trabajo las herramientas necesarias para desarrollar y experimentar. Los parámetros de control del son: los ángulos de cabeceo (pitch), balanceo (roll) y guiñada (yaw). El desarrollo se realizó en el lenguaje C++, utilizando una librería llamada libcflie [11] necesaria para conectarse con el drone (cuadricóptero), OpenCV [12] y un entorno Linux (Ubuntu 12.04). En la figura 6 se muestra una imagen del drone con un parche de color para su identificación por parte del sistema de visión artificial.

Figura 6. Cuadricóptero con identificación de color

Actualmente se trabaja con un robot ARDrone, de la empresa Parrot [13]. En la figura 7, se muestra la navegación sobre los checkpoint (parches que representan los objetivos de la misión)

Figura 7. ARDrone ejecutando el recorrido

3. Arquitectura del Sistema

La arquitectura del sistema desarrollado permite la carga de una librería encargada de la rutina de navegación, y otra librería para la mecánica de vuelo. Se provee además, un método de comunicación entre ambas librerías que implementa una lista de "temas de conversación", extensible a las necesidades de ambas. El software desarrollado crea un hilo de ejecución encargado del análisis del video, con el algoritmo presentado en [14] y [15], y publica los mensajes relacionados a la posición del robot y la posición de los checkpoints. A su vez, esa información la toma el hilo encargado de la interfaz gráfica, y el hilo de la rutina de navegación. Por otro lado, el algoritmo de la mecánica de vuelo [16] publica mensajes relacionados a la información proveniente del robot, como por ejemplo: altura, inclinación, velocidad, entre otros. En detalle, la rutina de navegación debe obtener la posición actual del robot y definir el próximo destino al que ir en base a los datos que provee el análisis del video y a la configuración de la misión a cumplir. En ella debe establecer tanto los destinos que debe cumplir, como la altura y el tiempo de sobrevuelo/aterrizaje de los mismos. Además, debe ser capaz de reconocer cuándo el objetivo se cumplió satisfactoriamente para ir al siguiente punto (objetivo representado por un parche de color), y por último, debe ser capaz de mantener al robot dentro de los límites del campo de la misión, ya que no existen paredes físicas. El algoritmo de navegación, debe ser capaz de recibir un punto destino y enviar al robot hacia allí en el menor tiempo posible, y mantenerlo a la altura correspondiente.

Para el algoritmo de navegación se utilizó un algoritmo híbrido presentado en [16]. Este algoritmo se definió al comparar los resultados de pruebas realizadas en donde la información de la velocidad del robot se calculaba comparando cuadros consecutivos del video obtenido por la cámara suspendida. En contraposición, el algoritmo híbrido, utiliza la información de la velocidad proveniente de los sensores del robot, de esta forma, se eliminan los posibles errores y dilaciones que pueda llegar a tener el cálculo de la velocidad utilizando el video, y que se propagarán al resto del algoritmo de control utilizado. Los resultados de las pruebas, mostraron una mejora de alrededor del 20% utilizando el modelo híbrido de navegación. Si bien esta diferencia la consideramos aceptable, nos inclinamos a utilizar este modelo dado su mayor confiabilidad.

En la figura 8 se presenta un diagrama de la arquitectura, y el pasaje de mensajes entre los hilos de ejecución.

El primer nivel de competencia establece la interpretación de la misión y la configuración de la misma. Este nivel se ve representado en la configuración de la rutina de vuelo, que alimentará a la misma con información sobre los objetivos a alcanzar

El segundo nivel, establece el desarrollo de la rutina de vuelo que se materializa en una librería utilizada por el software y por el algoritmo de la mecánica de vuelo. Este nivel, como se dijo anteriormente, está orientado a estudiantes de los primeros años de carreras de ingeniería en informática o afines, y busca la inserción de los mismos en el ámbito de la robótica.

Por último, el tercer nivel establece el desarrollo de la mecánica de vuelo materializado en una librería que se comunica con el robot (en este caso un ARDrone 2.0) y

define los valores utilizando algoritmos de control. Este nivel, está orientado a estudiantes de ingeniería avanzados (figura 8).

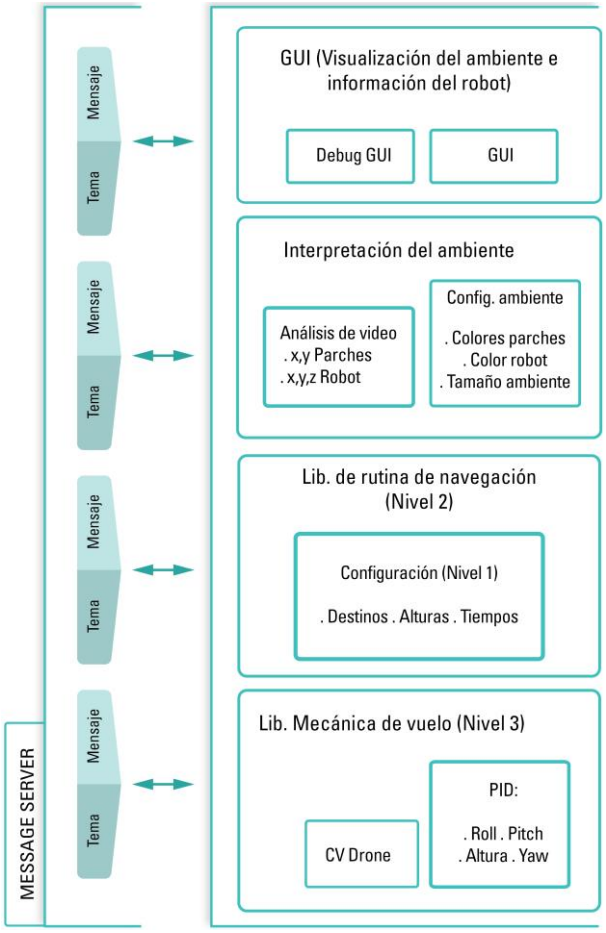

Figura 8. Arquitectura, y pasaje de mensajes entre los hilos de ejecución

4. Ambiente de competición y experimentación

El ambiente de competición y experimentación se desarrolla en tres niveles de participación de estudiantes, el primer nivel orientado a estudiantes de escuelas secundarias orientado a la interpretación de misiones; el segundo nivel orientado a estudiantes universitarios del primer ciclo de las carreras de informática; y el tercer nivel orientado a estudiantes de grado avanzados de las carreras de ingeniería informática. En el primer nivel de competición, los participantes deberán interpretar una misión que involucra distintos checkpoints o destinos que representan puntos de interés representados por los parches de colores, por ejemplo un incendio, una inundación, un rescate, etc. Este nivel está orientado a estudiantes de nivel secundario y apunta a la interpretación y configuración de una misión por ejemplo de búsqueda y rescate. De esta forma, el robot encargado de ejecutar la misión deberá alcanzar los destinos definidos en la configuración y sobrevolar durante un tiempo establecido la zona, considerando de corresponder su aterrizaje y despegue (por ejemplo: en una situación de rescate, recolección de agua, etc). Una vez definida la misión, los participantes del segundo nivel de competición, deberán programar una rutina de navegación, que enviará al robot a los destinos definidos, a la altura correspondiente y durante el tiempo necesario. Este nivel, está orientado a estudiantes de nivel universitario del primer ciclo, y pretende introducirlos en el desarrollo de algoritmos de navegación. En el tercer nivel los estudiantes de ingeniería en niveles avanzados pueden involucrarse, además de en el desarrollo de la navegación, en el control de la mecánica de vuelo del drone.

El sistema brinda un formato de competición del tipo "búsqueda y rescate" bajo la restricción del tiempo para el cumplimiento de la misión. Los equipos participantes reciben un texto con la misión, el que deben interpretar para poder definir el orden de los objetivos a cumplir (puntos a alcanzar, la altura que debe tomar el robot al alcanzar esos puntos y el tiempo que debe permanecer en el destino). El mejor equipo será aquel que logre realizar la misión en el menor tiempo posible, sin saltearse ningún punto y cumpliendo todas las restricciones del caso en función de la misión. Se establecen tres niveles de competición: el primero, orientado a la interpretación de la misión, esto es, leer el texto dado y definir los puntos a donde debe acercarse el robot, el tiempo y la altura; el segundo nivel, orientado a estudiantes con conocimientos básicos de control y programación, en el cual deberán programar la librería de la rutina de navegación con el objetivo de completar la misión. Los estudiantes participantes en el segundo nivel, con conocimientos de programación, deberán programar el control de navegación del drone y por último, el tercer nivel, orientado a estudiantes avanzados, con conocimientos de control y programación, en el que deberán programar el control del robot el algoritmo de navegación y la mecánica de vuelo. Como se ve en la figura 9, cada nivel incluye al anterior.

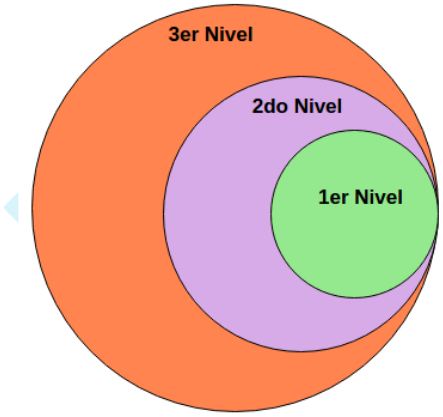

Figura 9. Gráfico con los niveles de dificultad. Los niveles más avanzados incluyen a los anteriores.

5. Resultados de las experiencias iniciales realizadas

Como forma de demostración de los resultados, se idearon 4 misiones de pruebas y se desarrolló una interfaz, que permite la configuración de las misiones, para el primer nivel de competición. Estos 3 niveles fueron presentados durante el Congreso TE&ET 2016, que se llevó a cabo en la Universidad de Morón.

5.1. Interfaz gráfica para el primer nivel de competición

Se desarrolló un sistema sencillo para realizar la configuración de las misiones que debe realizar el drone. Este sistema, presenta la misión de forma textual, y luego permite al participante/alumno definir la cantidad de destinos y establecer los parámetros de vuelo para cada uno de ellos. De este modo, el alumno con escasos conocimientos de computación o programación, puede participar de las competencias en el primer nivel utilizando este sistema (figura 10).

Figura 10. Herramienta de configuración de misiones.

Como resultado de la utilización de esta herramienta, el participante/alumno obtiene el archivo de configuración correspondiente a la misión dada. De esta forma, el trabajo de configuración se simplifica notablemente, y se centra el trabajo de interpretación de la misión.

El archivo resultante, es en formato JSON [17] y tiene la siguiente estructura:

```
{
"checkpoints":[
   { "id": 1, "altura": 1000, "time": 2 },
  ....
  ]
}
```

Aquí se puede observar una colección de checkpoints, donde se especifica un ID que corresponde al color en el ambiente, la altura en milimetros, y el tiempo durante el cual el robot debe sobrevolar el checkpoint, expresado en segundos.

5.2. Misiones de prueba

Las siguientes misiones fueron ideadas para la experimentación del sistema propuesto, en el contexto de las materias Robótica y Sistemas Inteligentes, de la carrera Ingeniería en Informática, y además se efectuaron demostraciones de éstas durante el Congreso TE&ET 2016. Se pensaron con la idea de mostrar distintas situaciones hipotéticas, con distintas restricciones como pueden ser la altura, o el tiempo que se debe sobrevolar un checkpoint. De esta forma, se puedo comprobar que el framework desarrollado permite cierta flexibilidad a la hora de idear una misión, y que además es lo suficientemente robusto para llevarlas a cabo. Por otro lado, no es importante la disposición de los parches en el ambiente, mientras que todos los competidores dispongan del mismo ambiente. Esto es así puesto que lo más importante para el desarrollo de las misiones, es que se respete la rutina de navegación, en el tiempo y con las restricciones presentes.

5.2.1. Misión 1. Derrame de petróleo

Un barco de una empresa de transporte de petróleo se accidentó cerca de la costa de Necochea y derramó 1 toneladas de crudo. El gobierno Nacional intimó a la empresa a limpiar el desastre y ésta utilizará un nuevo sistema de limpieza que utiliza cuadricópteros asistidos por imagen satelital en tiempo real. Los robots tienen una autonomía de 2 horas y en ese lapso de tiempo cada cuadricóptero alcanza a limpiar 10 kilogramos de crudo. Luego de realizar la limpieza, cada cuadricóptero debe volver a tierra a recargarse. El procedimiento de limpieza es el siguiente: los cuadricópteros salen de tierra hasta el derrame en el océano; durante 2 horas se encargan de mover, en 2 viajes, los 10 kilos de petróleo a un barco que se encuentra cerca del derrame. Para poder succionar el petróleo, los cuadricópteros deben permanecer a 1 metro de distancia y tardan 1 hora. Para la descarga deben estacionar, tardando 30 minutos en realizarla. Luego de los 10 viajes, el robot debe estar 2 horas recargándose en tierra. La empresa dispone de 200, por lo que cada uno de ellos debe remover 20 kilogramos de crudo.

Se debe realizar la misión teniendo en cuenta que:

- 1 hora corresponde a 2 segundos.
- 1 metro corresponde a 1 metro.
- El barco de petróleo está representado por el color marrón.

- La costa está representada por el color verde
- El derrame de petróleo está representado por el color azul.

Solución

El robot debe hacer 2 viajes completos, permanecer 4 segundos en tierra cuando tenga que recargar, 2 segundos por viaje arriba del parche azul y 1 segundo por viaje sobre el barco. La duración total de la misión es de 14 segundos.

En los videos siguientes se puede observar la ejecución de la misión: https://goo.gl/IHfFhP, https://goo.gl/rNs4o9, https://goo.gl/CEEoHe.

5.2.2. Misión 2. Terremoto en Mendoza.

Ha ocurrido un terremoto en una gran ciudad de la provincia de Mendoza, dejando centenares de personas atrapadas en un lugar inaccesible para un rescate normal. El gobierno provincial contrató a una empresa dedicada al rescate en lugares inaccesibles mediante la utilización de guiados por cuadricópteros situados a una gran altitud. De este modo, se provee una visión general de la situación y se guía al resto de los cuadricópteros para el rescate. La empresa cuenta con 100 cuadricópteros capaces de transportar a 1 persona cada uno. Se estima que la cantidad de personas atrapadas es 200, por lo que cada cuadricóptero deberá realizar 2 viajes. Por otro lado, la autonomía de cada uno alcanza para rescatar 1 persona, y el tiempo de recarga en la base es de 2 horas. Además, es necesario que los cuadricópteros en cada viaje dejen suministros para aquellos que no van a ser rescatados. En el primer viaje los cuadricópteros deben llevar agua, y en el segundo viaje comida, para realizar esto, cada cuadricóptero tarda 30 minutos en realizar la carga de agua sin estacionar, a una altura de 2 metros, y 30 minutos estacionando en el caso de la comida. El tiempo que tarda en cargar una persona es de 30 minutos y lo realiza a una altura de 1 metro.

Se debe realizar la misión teniendo en cuenta que:

- 1 hora corresponde a 2 segundos.
- 1 metro corresponde a 50 centímetros.
- La base está representada por el parche naranja.
- Las personas están representadas por el parche amarillo.
- El agua está representada por el parche celeste.
- La comida está representada por el parche verde

Solución

De la interpretación del texto por parte del alumno y su selección de parches para cargar la misión empleando la interfaz grafica correspondiente al primer nivel, surge que: el robot debe despegar y pasar a buscar agua en el parche celeste, a 1 metro, durante 1 segundo. Luego, debe ir al parche amarillo y quedarse 1 segundo a 50 centímetros. Debe volver al parche naranja, y estaciona durante 4 segundos. Luego va al parche verde, estaciona 1 segundo y por último va al parche amarillo, 1 segundo a 50 centímetros. Vuelve al parche naranja. La duración total de la misión es de 8 segundos.

El archivo resultante, es en formato JSON, tiene la siguiente estructura:

```
{ "checkpoints":[
    { "id":1, "altura": 1000, "time": 2 },
    { "id":2, "altura": 0, "time": 1 },
    { "id": 3, "altura": 0, "time": 4 },
    { "id":1, "altura": 1000, "time": 2 },
    { "id":2, "altura": 0, "time": 1 },
    { "id": 3, "altura": 0, "time": 4 }
  ]
}
```

En los siguientes videos se puede observar la ejecución de la misión: https://goo.gl/P45SsQ, https://goo.gl/joAiwg, https://goo.gl/FltQrY. (figura 11)

Figura 11. Robot realizando la misión 2.

5.2.3. Misión 3. SheepTacking soluciones ganaderas.

La empresa SheepTracking es una empresa dedicada a aplicar tecnología de vanguardia en el ámbito ganadero. Esta empresa ideó un sistema de seguimiento mediante la implantación de un identificador subcutáneo en animales de granja. Para automatizar la implantación del identificador, utiliza cuadricópteros autónomos guiados por otro cuadricóptero ubicado a gran altitud de modo de obtener e identificar objetivos de interés.

Como este sistema es nuevo, utilizará a un cliente para probarlo. Este cliente dispone de 100 vacas y 50 chanchos y desea realizar un seguimiento de todos ellos. Un cuadricóptero puede marcar 100 animales y coloca 10 implantes por minuto. Una vez marcados 100 animales, debe recargar los implantes en la base y para esto debe estacionar durante 10 minutos. Para las vacas el cuadricóptero debe situarse a 1.5 metros de altura, y para los chanchos, debe situarse a 50 cm de altura.

Se debe realizar la misión teniendo en cuenta que:

- minutos corresponden a 1 segundo.
- 1 metro corresponde a 1 metro.
- Las vacas están representadas por el color naranja.
- Los chanchos están representados por el color amarillo
- La base está representada por el color verde.

Solución

De la interpretación del texto por parte del alumno y su selección de parches para cargar la misión empleando la interfaz grafica correspondiente al primer nivel, surge que: el robot debe despegar e ir al parche naranja, situarse a 1,5 metros durante 5 segundos. Luego debe volver al parche verde, estacionar durante 5 segundos. Por último, debe ir al parche amarillo, situarse a 50 centímetros durante 3 segundos y volver al parche verde. En total la misión debe durar 13 segundos.

En los siguientes videos se puede observar la ejecución de la misión: https://goo.gl/BRQPnQ, https://goo.gl/sXnzu2, https://goo.gl/OaeZwa.

5.2.4. Escenario para la pruebas de algoritmos de control de vuelo y navegación

Para las pruebas / competición centradas en algoritmos de control de vuelo y navegación de cuadricópteros se diseñó un circuito de prueba asistido por una cámara suspendida a 3 metros de altura que guíe al robot. Este circuito cuenta con las siguientes características: son 4 parches en el suelo, y cada uno de ellos representa una altura determinada. El parche **A**, representa una altura de 50 centímetros y un tiempo de sobrevuelo de 2 segundos. El parche **B** representa una altura de 1 metro, y un tiempo de sobrevuelo de 4 segundos. El parche **C** representa una altura de 75 centímetros y un tiempo de sobrevuelo de 6 segundos. Por último, el parche **D**, representa una altura de 1.5 metros y un tiempo de sobrevuelo de 8 segundos. Una vez finalizado ese circuito, el robot debe volver al parche A y estacionar allí. El robot debe partir del centro de la figura. La disposición de los parches se puede observar en la en la figura 12.

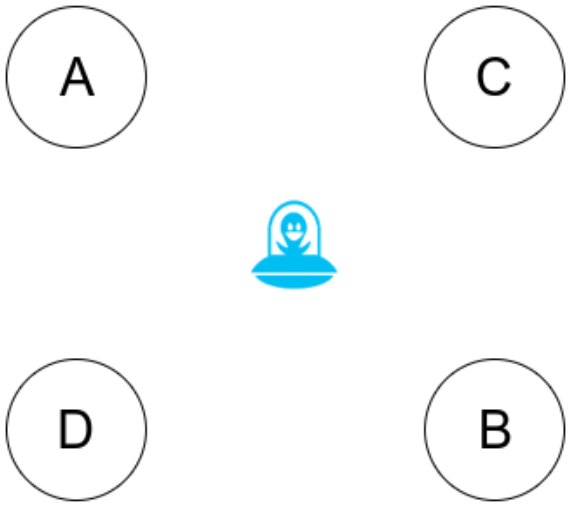

Figura 12. Disposición de parches misión 4.

Solución

De la interpretación del texto por parte del alumno y su selección de parches para cargar la misión empleando la interfaz grafica correspondiente al primer nivel, surge que: el robot debe despegar del centro del diagrama y dirigirse al parche A. Una vez allí, debe permanecer a 50 centímetros durante 2 segundos. Luego debe dirigirse al parche B, y permanecer durante 4 segundos a una altura de 1 metro. Luego, debe dirigirse al parche C y permanecer durante 6 segundos a una altura de 75 centímetros. Una vez logrado esto, debe ir al parche D y permanecer a una altura de 1.5 metros, durante 8 segundos. Para finalizar, debe dirigirse al parche A y estacionar allí. El tiempo total de la misión es de 20 segundos.

En los siguientes videos se puede observar la ejecución de la misión: https://goo.gl/cXlsz0, https://goo.gl/8IFYP2, https://goo.gl/sZl4ZU.

Estas misiones utilizaron las librerías desarrolladas para la rutina de navegación y para el control del robot (utilizando la navegación híbrida presentada en [16]), y se configuraron las misiones utilizando la herramienta de configuración.

6. Futuras líneas de trabajo

En la actualidad debe ser una persona o la propia rutina de navegación la que se encargue de dos cosas: por un lado, el informar al robot que se ha salido de los límites de la visión de la cámara y por otro, establecer si se cumplió o no un hito de la misión. Para solucionar esto, se propone en un futuro, desarrollar un sistema de monitoreo, tanto para la seguridad del robot como para arbitrar el cumplimiento de la misión.

6.1. Arbitraje automático

Se pretende modificar la aplicación para que sea quien establezca si el recorrido se realizó de manera correcta, teniendo en cuenta las alturas, los tiempos y los parches por donde pasó el robot.

Un caso de ejemplo es, si un parche representa fuego el robot no podrá volar por una altura más baja que la indicada, o más tiempo que el establecido; y si la misión era apagar el fuego, primero deberá pasar por el parche que represente agua para luego apagar el fuego. Algunos de estos parámetros mencionados pueden ser difíciles de medir por una persona en tiempo de vuelo del robot, es por esto que un árbitro automático hará transparente la evaluación del ganador.

6.2. Monitor de seguridad

El robot debe estar siempre en el campo de visión de la cámara y si en algún momento se va del mismo, la rutina debería finalizarse, terminando así la competencia. Para que esto no suceda se deben tomar acciones de control necesarias para llevar al robot nuevamente al campo de visión.

Como se mencionó en las secciones anteriores, en la actualidad estas acciones de control quedan en manos del competidor.

En un futuro se pretende que exista un monitor de seguridad, es decir que estas directivas de control sean homogéneas para todos los participantes. De esta forma se logra una competencia dinámica, ya que estas acciones son de gran importancia para lograr un recorrido fluido.

6.3. Implementación en materias universitarias y secundarias.

Una parte importante de la demostración del framework es su utilización en alguna materia de la carrera Ingeniería en Informática, o en talleres de robótica educativa en escuelas secundarias de la zona.

Dentro de la universidad, es posible incentivar a los alumnos, e introducirlos en la investigación en robótica a partir de materias como Sistemas Inteligentes (alumnos de 5to año de la carrera), o Programación 2 y 3 (alumnos de 1er y 2do año). Los alumnos de los primeros años de la carrera podrán trabajar en los primeros dos niveles de competición, y los alumnos de los últimos años podrán hacerlo en el tercer nivel.

La introducción de competencias a nivel universitario, permitirá demostrar no sólo el funcionamiento del framework, sino que también permitirá incentivar a los alumnos en la investigación de sistemas inteligentes y de tiempo real.

Conclusiones

La implementación de competencias en educación conlleva la generación de un ambiente tanto lúdico, como de experimentación y aprendizaje, incentivando a los alumnos a través del juego.

Los diferentes niveles de aprendizaje que se presentaron en este artículo llevan a que puedan involucrarse en la competencia tanto alumnos recién iniciados, como alumnos avanzados, abarcando así diferentes niveles de conocimiento.

Por otro lado, se pone a disposición un ambiente de experimentación para algoritmos de navegación que permite evaluar la precisión y velocidad del mismo, permitiendo comparar un algoritmo con otro en términos de navegación como de control de la mecánica de vuelo

Durante el congreso TE&ET 2016 en el marco de las demos de educación en robótica, se demostró al público la utilización del framework desarrollado. Allí se expuso su funcionamiento. En la figura 13 se puede observar a dos integrantes exponiendo.

Figura 13. Exposición en el congreso TE&ET 2016.

Referencias

[1] RoboCup, http://goo.gl/dRZFes

[2] FIRA, http://goo.gl/AH127O

[3] Robot Challenge, https://goo.gl/SM0SsV

[4] FIRA MiroSot, http://goo.gl/sFAl26

[5] FIRA NaroSot, http://goo.gl/FqblDV

[6] FIRA RoboSot, http://goo.gl/k1W2Ui

[7] Small Size League, http://goo.gl/H95BY1

[8] IARC, https://goo.gl/Lx0t7e

[9] SUAS, https://goo.gl/XUWLzE

[10] Bitcarze AB, http://www.bitcraze.se/

[11] Crazyflie Nano C++ Client Library, https://github.com/fairlight1337/libcflie

[12] OpenCV, http://opencv.org/

[13] ARDrone 2.0 de Parrot, http://ardrone2.parrot.com

[14] Avila, Fasce, Lorusso, Ierache. Robótica Situada Aplicada al Control de Vehículos Aéreos Autónomos. V Workshop Procesamiento de Señales y Sistemas de Tiempo Real, XX Congreso Argentino de Ciencias de la Computación, Octubre 2014 ISBN 978-987-3806-05-6.

[15] Lorusso, Emiliano; Fasce, Sofia; Ávila, Diego; Pereira, Gustavo; Ierache, Jorge Salvador. Experiencias en el control de un Drone en el contexto de la robótica situada. Jornadas Argentinas de Tecnología, Innovación y Creatividad JATIC Mar del Plata Noviembre 2015 ISBN 978-987-23963-2-9.

[16] Fasce Sofía; Lorusso Emiliano,Avila Diego ; Ierache Jorge, Pereira Gustavo, Autonomous Control of a Drone in the Context of Situated Robotics. ISSN 2194-5357 Advances in Intelligent Systems and Computing.: Springer International Publishing. 2016 vol.447 n°. p185 – 194

[17] JSON, http://www.json.org

Información de Contacto de los Autores

Jorge Ierache
Cabildo 134
Morón, Buenos Aires
Argentina
jierache@unimoron.edu.ar
http://www.unimoron.edu.ar

Diego Avila
Investigador-Alumno del Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica, Universidad de Morón.

Emiliano Lorusso
Investigador-Alumno del Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica, Universidad de Morón.

Sofía Fasce
Investigador-Alumno del Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica, Universidad de Morón.

Jorge Ierache
Dr. en Ciencias Informáticas, Director del Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica, Universidad de Morón.