

USO DE SIMULADORES GAZEBO Y SITL (SOFTWARE IN THE LOOP) PARA VEHÍCULOS AÉREOS NO TRIPULADOS EN ENTORNOS VIRTUALES 3D

M. en C. Esther Viridiana Vázquez Carmona
Centro de Innovación y Desarrollo Tecnológico en
Cómputo
evazquezc1801@alumno.ipn.mx
M. en C. Luis Alberto Flores Montaña
Centro de Innovación y Desarrollo Tecnológico en
Cómputo
luisfloresmontano@hotmail.com
M. en C. Rodrigo Vázquez López
Centro de Innovación y Desarrollo Tecnológico en
Cómputo
rvazquezl1800@alumno.ipn.mx
Dr. Juan Carlos Herrera Lozada
Centro de Innovación y Desarrollo Tecnológico en
Cómputo
jlozada@ipn.mx

Instituto Politécnico Nacional

Resumen

Debido a la demanda de nuevos sistemas en el ámbito de la industria, la robótica se ha convertido en un tema importante de estudio y desarrollo dentro del campo de la investigación. La robótica hace uso de diferentes disciplinas, como la mecánica, electrónica, inteligencia artificial, física y e ingeniería de control, entre otras. Su objetivo es construir sistemas que puedan llevar a cabo tareas de usuario en las que se requiere la intervención humana o automatización, siendo a su vez capaces de operar en entornos inestables y poseer un mayor grado de autonomía. Para ello es importante mencionar que dentro de las etapas que integran la metodología del desarrollo de dichos sistemas, se encuentra la etapa de simulación de realidad virtual, la cual permite al robot funcionar en un entorno virtual, con características muy similares a las de los ambientes reales para no arriesgar equipos, vidas obteniendo resultados similares a los que se tendría en la vida real. En este documento se muestra el uso del simulador Gazebo en colaboración con SITL para ejecutar una trayectoria de un vehículo aéreo no tripulado (VANT) dentro de un entorno 3D predefinido por Gazebo. Adicionalmente se utiliza el software de Mission Planner de Ardupilot que se encarga de generar los waypoints o coordenadas dentro de una región de interés (en inglés Region of Interest ROI) para realizar el recorrido del VANT dentro de dicha región. Se anexan las figuras que integran la simulación del VANT, así como las configuraciones y comandos necesarios para llevar a cabo la misión.

1. Introducción

El desarrollo de sistemas robóticos requiere de diseño y construcción de robots que puedan realizar trabajos forzados o realizar tareas en el que el ser humano pueda estar en peligro, como manipulación de residuos químicos, inspección o mantenimiento (Revista de Robots, 2021)

1.1 Automatización y robótica

Los últimos avances de la ingeniería robótica han demostrado que los robots pueden realizar tareas complejas como manipulación de objetos, realizar operaciones de forma autónoma o tareas que

requieren de colaboración. Por ejemplo, para un sistema automatizado, las producciones en masa y a gran escala resultan ser productivas debido a la precisión, también existen robots que pueden ser reprogramables para ejecutar diversas tareas o robots que se encuentran en diferentes etapas de un proceso, como la obtención de información, el procesamiento y la toma de decisiones. Para llevar a cabo estas tareas, es necesario establecer funciones en cada subsistema del robot, esto se le conoce como metodología de diseño de un robot. La principal característica de esta metodología es que está integrada por diferentes disciplinas como el diseño conceptual, el análisis cinemático, el análisis dinámico, el diseño mecánico y la simulación virtual que se ejecutan de forma paralela (Kchir, 2016).

1.2 Simuladores

Los simuladores son entornos que permiten trabajar la programación de los módulos antes mencionados de manera virtual, emulando cada una de sus funciones, algunos permiten la simulación 3D en ambientes realistas con el objetivo de investigar el comportamiento de los robots móviles sin necesidad de construirlo, garantizando la realización de pruebas a muy bajo costo. MATLAB y Simulink, Fleet Simulator, EV3DEV Python Simulator, Miranda, mBot robot simulator, Codey Rocky simulator, Open Roberta Lab, Robot Virtual Worlds, Gazebo, SITL, FlightGear, GEoFS, Google Earth, Rise of Flight, World of Warplanes son solo algunos simuladores para el diseño de sistemas de control y navegación de vehículos terrestres y aéreos. Para este trabajo de investigación se eligió el simulador Gazebo debido a que puede simular robots en entornos exteriores 3D bajo diversas condiciones compatible con el uso independiente (sin ROS) en colaboración con SITL que permite probar el comportamiento del vehículo sin uso del hardware, en este del propio vehículo (JuegosRobotica, 2020).

1.3 Mission Planner de Ardupilot

Ardupilot es una plataforma conformada por varias herramientas que permite el control, la comunicación en tiempo real y el uso de sistemas en VANTS. Una de las herramientas de esta plataforma es Mission planner, que permite planear una misión de vuelo a través de puntos de referencia, coordenadas o waypoints, para ello se dibuja la forma de un polígono sobre la ROI, posteriormente se crea una serie de waypoints para que el vehículo realice la trayectoria. Para llevar a cabo la comunicación entre Gazebo y SITL, se utiliza el protocolo MAVLink, que permite el intercambio de mensajes entre una estación de control en tierra (en inglés ground control station GCS) y el VANT, en este caso MAVProxy trabaja como un GCS que admite el protocolo MAVLink, está basado en comandos y puede trabajar con Mission Planner para proporcionar una interfaz gráfica. En la tabla 1 se presentan los comandos más utilizados de MAVProxy, así como la función que cada uno realiza (Ardupilot, 2021).

Tabla. 1.
Comandos MAVproxy.

Comando MAVProxy	Función
arm throttle	Permite activar el piloto automático del VANT para armar y desarmar de forma remota.
mode guided	Permite elegir el modo de vuelo del VANT, vuela hasta el último punto en el que se hizo clic en el mapa, o la longitud y latitud especificadas.
mode auto	Permite elegir el modo de vuelo del VANT, comenzando por el primer punto del waypoint.
takeoff + altura	Permite despegar el VANT en una misión a una cierta altura.
wp load filename.txt	Permite cargar un nuevo conjunto de waypoints desde archivo para el VANT.
wp loop	Cierra el ciclo de una misión, lo que permite que el VANT repita una misión.
wp set + número de waypoint	La misión se ejecutará a partir de este punto de partida.

2. Desarrollo

Se realizó la misión en un VANT sobre una ROI, a través del simulador Gazebo en colaboración con SITL. Para ello se eligió una ROI a los alrededores del Bosque de Chapultepec que puede llegar a utilizarse con fines de monitoreo o inspección.

2.1 Delimitación de la ROI y waypoints en Mission Planner

Para realizar la misión en un VANT, se utilizó el software Mission Planner, para delimitar el área de interés, primero se insertan las coordenadas (latitud, longitud y altitud) donde el VANT realizará el despegue, también llamado home. Las coordenadas utilizadas, se ubican en el castillo de Chapultepec, posteriormente se delimita la región, dibujando un polígono alrededor del área. En la figura 1 se puede observar la ROI delimitada por un polígono, así como los parámetros antes mencionados.



Figura 1. . Las coordenadas ingresadas para la ROI fueron 19.4201704705298, -99.1819149255753, 25.

Posteriormente se generó el archivo de waypoints con Mission Planner, a través de la función Auto WP que se encuentra en el menú, este archivo puede tener como extensión .txt o .waypoints. En la figura 2 se puede observar la ruta o path generado.

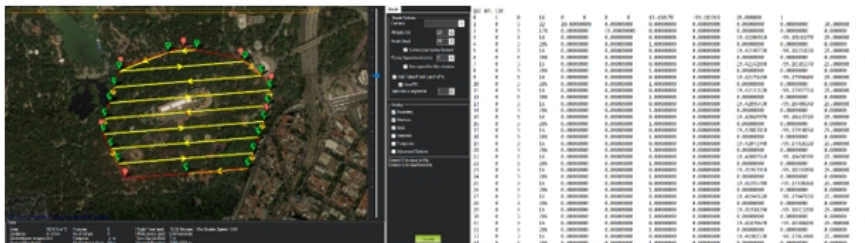


Figura 2. Path generado por Mission Planner, en la parte lateral pueden modificarse parámetros como altitud, ángulo de giro, la velocidad, la distancia entre líneas de vuelo, entre otros que se utilizarán durante la misión del VANT. Una vez que se modificaron estos parámetros, los waypoints se guardan en un archivo, el cual se puede observar en la figura de la derecha.

2.2 SITL

El simulador SITL suele ejecutarse en Linux y en Windows, debido a que existen algunos paquetes ya obsoletos que deben ocuparse para ambos simuladores, se trabajó sobre la distribución de Linux en Ubuntu 18. En la figura 3 se muestra el proceso de instalación (ArduPilot Dev,2021).

```
Archivo Editar Ver Buscar Terminal Ayuda
viry@viry-Insprion-5559:~$ sudo apt-get update
[sudo] contraseña para viry:
Obj:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Obj:2 http://archive.canonical.com/ubuntu bionic InRelease
Obj:3 http://mx.archive.ubuntu.com/ubuntu bionic InRelease
viry@viry-Insprion-5559:~$ sudo apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
git ya está en su versión más reciente (1:2.17.1-subuntu.8).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
libappindicator1 libindicator7 linux-headers-5.4.0-42-generic linux-hwe-5.4-headers-5.4.0-42 linux-image-5.4.0-42-generic
linux-modules-5.4.0-42-generic linux-modules-extra-5.4.0-42-generic python3-click python3-colorama
utilscc +sudo apt autoremove para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 27 no actualizados.
viry@viry-Insprion-5559:~$ sudo apt-get install gitk git-gui
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
viry@viry-Insprion-5559:~$ git clone https://github.com/ArduPilot/ardupilot.git
Clonando en 'ardupilot'...
remote: Enumerating objects: 337432, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (118/118), done.
remote: Total 337432 (delta 49), reused 24 (delta 8), pack-reused 337314
Receiving objects: 100% (337432/337432), 218.07 MiB | 4.26 MiB/s, listo.
Resolviendo deltas: 100% (24055/24055), listo.
viry@viry-Insprion-5559:~$ cd ardupilot
viry@viry-Insprion-5559:~/ardupilot$ git submodule update --init --recursive
viry@viry-Insprion-5559:~/ardupilot$ Tools/environment_install/install-prereqs-ubuntu.sh -y
viry@viry-Insprion-5559:~/ardupilot$ ./.profile
viry@viry-Insprion-5559:~/ardupilot$ rmrf clean
```

Figura 3. Pasos para seguir para la instalación de SITL. Se crea una carpeta llamada ardupilot, en ella se encuentran los archivos con los parámetros para que el VANT pueda despegar.

Adicionalmente se debe modificar un archivo titulado “locations.txt” que se encuentra en ~ardupilot/Tools/autotest, en él se debe ingresar las coordenadas antes mencionadas de donde despegará el VANT y el nombre que caracteriza este lugar, para efectos de este trabajo lo nombraremos “Castillo= 19.4201704705298, -99.1819149255753, 25”.

2.3 Gazebo

Gazebo se caracteriza por ser un simulador externo para SITL. Además del simulador, se instala un complemento para prescindir de ROS llamado “khancyr” (Open Source Robotics Foundation, 2014) . A continuación, se presenta el proceso de instalación de Gazebo y el respectivo complemento, el cual se puede observar en la figura 4.

```
Archivo Editar Ver Buscar Terminal Ayuda
viry@viry-Insprion-5559:~$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable lib-release -cs main" > /etc/apt/sources.list.d/gazebo-stable.list'
[sudo] contraseña para viry:
viry@viry-Insprion-5559:~$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add
viry@viry-Insprion-5559:~$ sudo apt-get update
Resolviendo packages.osrfoundation.org (packages.osrfoundation.org)[52.52.171.73]
Conectando con packages.osrfoundation.org (packages.osrfoundation.org)[52.52.171.73]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1772 (1.7K) [application/octet-stream]
Guardando como: "STDOUT"
-
100K[=====] 1.73K --KB/s en 0s
2021-06-29 23:28:05 (53.8 MB/s) - escritos a stdout [1772/1772]
OK
viry@viry-Insprion-5559:~$ sudo apt update
viry@viry-Insprion-5559:~$ sudo apt install gazebo8 libgazebo8-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
fontconfig fonts-lato libappindicator1 libccid-dev libccid2 libgraphviz-dev
libjpeg-plugins-gtk libindicator7 libnbln-glib libnbln-proxy libopenal-dev
viry@viry-Insprion-5559:~$ gazebo --verbose
gazebo: symbol lookup error: /usr/lib/x86_64-linux-gnu/libgazebo_common.so.8: undefined symbol: _ZN8IgnitionIffuel_tools12ClientConfig12SetUse
/vAgentRKN0T7_cxx112basic_stringISt11Char_traitsSt11CharE
viry@viry-Insprion-5559:~$ git clone https://github.com/khancyr/ardupilot_gazebo
remote: Enumerating objects: 170, done.
remote: Total 176 (delta 8), reused 0 (delta 8), pack-reused 176
Receiving objects: 100% (176/176), 5.30 MiB | 1.70 MiB/s, listo.
Resolviendo deltas: 100% (79/79), listo.
viry@viry-Insprion-5559:~$ cd ardupilot_gazebo
viry@viry-Insprion-5559:~/ardupilot_gazebo$ mkdir -p build
viry@viry-Insprion-5559:~/ardupilot_gazebo$ cd build
viry@viry-Insprion-5559:~/ardupilot_gazebo/build$ cmake --
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/lib/ccache/cc
viry@viry-Insprion-5559:~/ardupilot_gazebo/build$ make -j4
Scanning dependencies of target ArduCopter2BlockPlugin
[ 55%] Building CXX object CMakeFiles/ArduCopter2BlockPlugin.dir/src/ArduCopter2BlockPlugin.cc.o
[ 50%] Building CXX object CMakeFiles/ArduPilotPlugin.dir/src/ArduPilotPlugin.cc.o
[ 75%] Linking CXX shared library libArduPilotPlugin.so
[ 75%] Built target ArduPilotPlugin
[100%] Linking CXX shared library libArduCopter2BlockPlugin.so
[100%] Built target ArduCopter2BlockPlugin
viry@viry-Insprion-5559:~/ardupilot_gazebo/build$ sudo make install
gazebo: symbol lookup error: /usr/lib/x86_64-linux-gnu/libgazebo_common.so.8: undefined symbol: _ZN8IgnitionIffuel_tools12ClientConfig12SetUse
```

Figura 4. Pasos para seguir para la instalación de Gazebo. El simulador y SITL se deben instalar en Ubuntu 16.04 o 18.04 ya que el complemento funciona solo con las versiones de Gazebo 7 o 9.

Para iniciar ambos simuladores, requiere insertar la instrucción `gazebo --verbose worlds/bayland_iris.world` en la terminal de Ubuntu. Cabe destacar que `bayland_iris.world` es un archivo tipo `world`, diseñado con anterioridad para modelar un entorno similar a un bosque. Se utiliza el mundo `bayland` que está dentro de los archivos de `gazebo` en combinación con el Cuadricóptero 3DR IRIS.

Posteriormente en otra ventana de la terminal se `cd ~/ardupilot/ArduCopter` seguido de `../Tools/autotest/sim_vehicle.py -L Castillo -f gazebo-iris --map --console`. La figura 5 muestra el entorno en el que se incluyen ambos simuladores.

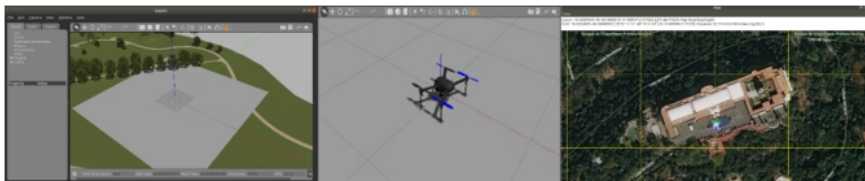


Figura 5. La parte izquierda de la figura muestra el ambiente de Gazebo, mostrado la vista exterior. La parte media de la figura muestra el ambiente de Gazebo haciendo un acercamiento al VANT. Finalmente, la parte derecha muestra el entorno de SITL.

3. Resultados

La simulación del VANT sobre el bosque de Chapultepec se llevó a cabo en un tiempo aproximado de 20min a una velocidad de 2m/s. En la figura 6 se muestran los comandos de MAVProxy para iniciar el recorrido de VANT.

c

```
vir@viry-B450-GAMING-X1:~$ gazebo --verbose worlds/bayland_iris.world
gazebo multi-robot simulator, version 9.19.0
Copyright (C) 2012 Open Source Robotics Foundation.
Released under the Apache 2 license.
http://gazebo.in.org

vir@viry-B450-GAMING-X1:~$ cd ~/ardupilot/ArduCopter
vir@viry-B450-GAMING-X1:~/ardupilot/ArduCopter$ ../Tools/autotest/sim_vehicle.py -L Castillo -f gazebo-iris --map --console
SIM_VEHICLE: Start
SIM_VEHICLE: Killing tasks
SIM_VEHICLE: Starting up at [19.419996, -99.183031, 40.0, 0.0] (Castillo)
SIM_VEHICLE: MAP build
SIM_VEHICLE: Configure waf
GUIDED> mode guided
GUIDED> takeoff 40
GUIDED> take off started

GUIDED> wp load /home/viry/ardupilot/Tools/autotest/Generis_Missions/Chapultepec.waypoints
GUIDED> Loaded 39 waypoints from /home/viry/ardupilot/Tools/autotest/Generis_Missions/Chapultepec.waypoints
wp set 1
GUIDED> auto
```

Figura 6. La figura muestra los comandos para iniciar el simulador de Gazebo y SITL, como se puede observar cada uno inicia en diferentes ventanas. Los comandos MAVProxy se introducen dentro de la terminal donde inicia SITL.

Finalmente, parte de la simulación se muestra en la figura 7.



Figura 7. En la parte izquierda de la figura, se muestra el entorno 3D. La parte media de la figura muestra la vista del VANT. Finalmente se puede observar como el VANT llega a la zona de aterrizaje sin chocar con los árboles, debido a la altura.

V. Conclusiones

En este trabajo, se muestra el uso del simulador Gazebo en colaboración con SITL. Se presentan los pasos a seguir para realizar la simulación de un VANT que seguirá una trayectoria sobre una ROI, en este caso se eligió el castillo de Chapultepec como punto de partida. Para ello se dibuja un polígono alrededor de la ROI, posteriormente Mission Planner se encarga de generar la trayectoria para que el VANT pueda realizar la misión siguiendo un conjunto de waypoints. Los waypoints son leídos por SITL, integra un GCS y a través de MAVProxy puede comunicarse con el simulador Gazebo, de modo que conforme el vehículo se mueva en SITL se vea reflejado en un entorno 3D que en este caso es el simulador Gazebo, con el objetivo de observar el comportamiento del VANT en un ambiente cercano a la realidad para fines de inspección, monitoreo y entrega de insumos o suministros.

Referencias

1. Ardupilot. (2021). *Mission Planner Home — Mission Planner documentation. Mission Planner Home*. Recuperado 20 junio, 2021, de <https://ardupilot.org/planner/>
2. ArduPilot Dev Team. (2021). *Ardupilot.Using Gazebo Simulator with SITL*. Recuperado 25 junio, 2021, de <https://ardupilot.org/dev/docs/using-gazebo-simulator-with-sitl.html>
3. JuegosRobotica. (2020). *Juegos Robótica. Robótica educativa #45 Entornos virtuales para programar robots emulados*. Recuperado 23 junio, 2021, de <https://juegosrobotica.es/podcast-045/>
4. AutorKchir, S., Dhouib, S., Tatibouet, J., Gradousoff, B., & da Silva Simoes, M. (2016). *RobotML for industrial robots: Design and simulation of manipulation scenarios. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Published. <https://doi.org/10.1109/etfa.2016.7733727>
5. Open Source Robotics Foundation. (2014). *Gazebo Tutorials. Gazebo. web* Recuperado 27 junio, 2021, de <http://gazebosim.org/tutorials>