

SIMULACIÓN DEL CONTROL Y LA COORDINACIÓN DE UN ROBOT EXPLORADOR EN UN AMBIENTE AGRÍCOLA

Marisol Rodríguez Cuevas

Universidad Nacional de Colombia

marrodriguezcue@unal.edu.co

Andrés Fernando Jiménez López

Universidad Nacional de Colombia

afjimenezlo@unal.edu.co

Pedro F. Cárdenas H.

Universidad Nacional de Colombia

pfcardenash@unal.edu.co

Resumen

En este artículo se presenta el procedimiento y los conceptos necesarios para la simulación de un vehículo terrestre no tripulado, con aplicaciones en el monitoreo de características fenológicas de cultivos agrícolas. La simulación se realiza mediante el uso de Gazebo a través de una máquina virtual comandada desde Matlab. El sistema de guiado del vehículo y detección se fundamenta en la visión artificial, que permite la definición de trayectorias específicas en un espacio abierto y la adquisición de imágenes de las plantas para estudiar su estado, como herramienta útil en la agricultura de precisión. El sistema busca ser la base tecnológica para el desarrollo de sistemas robóticos en campo abierto para la preparación del terreno, detección de enfermedades o estado nutricional de cultivos, facilitando el ingreso a los lotes, sin perjudicar el correcto desarrollo de las plantas.

Palabras Claves: Cultivos, Gazebo, jackal, máquina virtual, ROS.

Abstract

This paper presents the procedures and concepts necessary for the simulation of an unmanned land vehicle, with applications in the monitoring of phenologic

characteristics of agricultural crops. The simulation was done using Gazebo through a virtual machine commanded from Matlab. The guidance system of the vehicle and detection are based on artificial vision, which allows the definition of specific trajectories in an open space and the image acquisition of plants, to study their status; as a useful tool in precision agriculture applications. The system seeks to be the technological base for the development of robotic systems in open terrain for soil preparation, disease and weeds detection or nutritional status of crops, facilitating the entrance to the lots, without harming the correct development of plants.

Keywords: *Crops, Gazebo, jackal, ROS, virtual machine.*

1. Introducción

La agricultura de precisión (AP) es una estrategia holística, que busca la aplicación de insumos en el lugar, cantidad e instante adecuados. Uno de los sistemas más usados en AP son los sistemas robóticos, que permiten el seguimiento de trayectorias y la adquisición de información en campo, útil para la definición de requerimientos de insumos en los cultivos, que van desde la preparación del terreno hasta la obtención de la cosecha. La simulación de agentes físicos en entornos virtuales ha permitido el desarrollo de aplicaciones de prototipado y validación de respuestas de sistemas robóticos sin la necesidad de arriesgar la integridad física de los elementos que intervendrían en el mundo real, como es el caso de la agricultura. Mediante esta tecnología se puede adquirir información del desempeño de una máquina o algoritmo de manera eficiente, reduciendo en gran medida costos de investigación y desarrollo [Haslina, 2010]

Los escenarios robóticos son un campo de acción muy estudiado. Por medio de una simulación, un agente robótico puede interactuar con otros objetos y agentes en el mundo virtual, gracias a los sensores y actuadores que poseen. La complejidad del escenario virtual depende de la herramienta de software empleada, permitiendo, por tanto, el diseño de robots con diferentes funcionalidades [Vaughan, 2015].

Existen simuladores simples para robots sencillos hasta complejos simuladores en

3D, entre los que están: Stage, RViz y Gazebo. El primero es un simulador multi-robot que muestra al usuario un mundo definido en un archivo con una extensión *.world*. Este mundo contiene todo lo que el software necesita saber de los obstáculos, espacios libres con otros robots y objetos. RViz, como herramienta de simulador 3D de datos de otros nodos de ROS, permite incluso obtener datos de un robot real. Finalmente, Gazebo que es el simulador utilizado en este trabajo, simula el comportamiento de un robot en un mundo virtual, lo cual lo diferencia respecto a RViz. Gazebo permite, entre muchas opciones, diseñar robots de forma personalizada, crear mundos virtuales usando sencillas herramientas CAD e importar modelos ya creados [Francisco, 2005]. Además, es posible sincronizarlo con ROS de forma que los robots emulados publiquen la información de sus sensores en nodos, así como implementar una lógica y un control que dé ordenes al robot. Matlab juega un papel importante en este desarrollo porque permite simplificar y entender de una mejor manera cómo se ejecuta un nodo, cómo publicar o suscribirse, obteniendo de manera inmediata los resultados en el simulador. Además, porque si bien se desea hacer un proceso de investigación y desarrollo de procesos, para la realización de componentes matemáticos intensivos tales como instrumentos flexibles, cinemática inversa para robots, análisis de espacios de trabajo y diseño de control, requiere el uso de un ambiente de prototipado: Matlab/Simulink [Koubba, 2016].

ROS (Robotic Operative System por sus siglas en inglés) es un framework para el desarrollo de software para robots, que funciona como una interfaz del robot y combinando con Gazebo, resulta un simulador robótico robusto [Koenig, 2004]. ROS presenta funcionalidades de abstracción del hardware, manejo de componentes de bajo nivel, desarrollo de funcionalidades de uso común, comunicación de mensajes entre procesos y mantenimiento de paquetes. Se basa en una arquitectura de grafos, en los nodos se realiza procesamientos de recepción, envío y multiplexación de mensajes que provienen de sensores, actuación, control, estados y planificación.

Este artículo aborda el procedimiento para simular un ambiente y una aplicación de robótica en agricultura mediante la integración de Gazebo, Matlab y ROS.

2. Métodos

Gazebo

Gazebo es un simulador 3D soportado por el Open Source Robotics Foundation (OSRF). En la figura 1 se puede ver un robot Turtlebot esperando a recibir comandos estando rodeado por objetos proporcionados por Gazebo tales como armarios, paredes, cubos y cilindros.

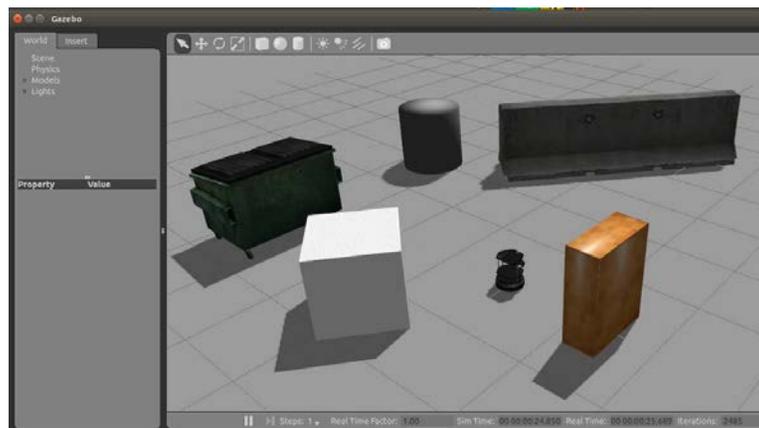


Figura 1 Turtlebot esperando comandos rodeado por objetos proporcionados por gazebo.

Gazebo fue diseñado originalmente para proporcionar simulaciones multirobot en un ambiente de exteriores de tres dimensiones. Fue concebido hacia finales del año 2002 en la Universidad del Sur de California por Andrew Howard y Nate Koenig como un simulador y aumentar las habilidades de Stage [Vaughan, 2015], un simulador ya existente diseñado para simulaciones 2D en ambientes interiores. Gazebo está disponible con licencia open-source y, así como muchos otros simuladores, usa ODE como su motor físico [Koenig, 2004]. En la figura 2 se muestra el esquema general con la arquitectura de Gazebo. Debido a que Gazebo fue diseñado para reflejar el comportamiento equivalente a la realidad, el software cliente usa una interfaz que se ve muy similar a la del robot real. El simulador es capaz de generar situaciones físicas tales como como la interacción a través de objetos (incluyendo una simulación precisa física de cuerpo rígido). Gazebo usa OpenGL (OpenGL 2010) para representar las imágenes obtenidas de las cámaras en simulación o simplemente como una herramienta de visualización. GLUT

(OpenGL Utility Toolkit) es una herramienta basada en OpenGL usada en Gazebo para la visualización de simulaciones. Las razones para adoptar estas herramientas en Gazebo son por la facilidad de uso, carga baja computacional independientemente en la plataforma [Koeing, 2004]. Las principales ventajas de Gazebo son:

- Simulación de diferentes sensores como sonares, láser escáner y GPS.
- Usa modelos comúnmente empleados, tales como Pioneer2DX, Pioneer2AT y SegwayRMP
- Soporte para plugins y programación en la nube.
- Motor de renderizado avanzado

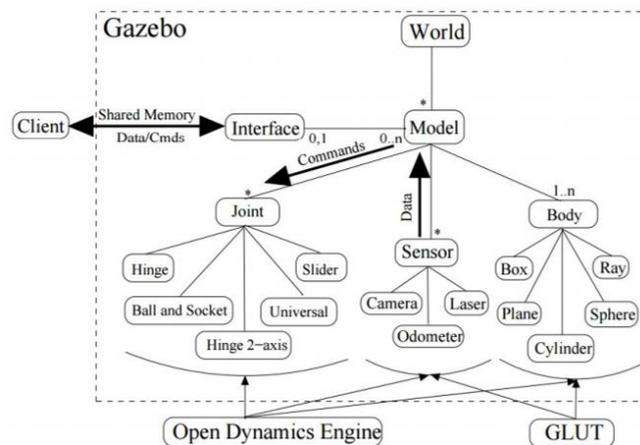


Figura 2 Esquema general con la arquitectura de Gazebo [Nathan, 2004].

Gazebo puede ejecutarse como simulador stand-alone, pero aquí se usará conjuntamente con Matlab y ROS para poder consultar desde los nodos que sean creados y la información de los sensores, así como para mandar comandos de velocidad entre otros.

Máquina Física

Gazebo emplea ODE (Open Dynamics Engine), una máquina o motor físico open source creado por Russel Smith, para simular la dinámica y cinemática asociada a las articulaciones de cuerpos rígidos. Incluyendo varias características como múltiples juntas, detección de colisiones, funciones de masa y rotación y muchas

geometrías, incluyendo mallas triangulares, ver figura 3 Gazebo incluye una capa entre lo que proporciona ODE y los modelos de Gazebo. Esta capa permite la creación de objetos tales como rayos láser y planos de tierra, manteniendo la funcionalidad que brinda ODE.

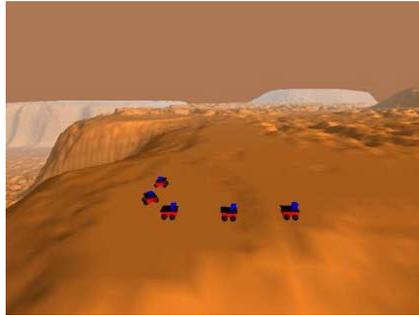


Figura 3 Infraestructura para conectar Gazebo

Visualización

OpenGL y GLUT son las herramientas por defecto para visualización. OpenGL es una librería estándar para la creación de aplicaciones interactivas 2D y 3D. Es una plataforma independiente, altamente escalable, estable y continuamente está evolucionando. GLUT es un toolkit para OpenGL. Las escenas hechas usando OpenGL se muestran en ventanas creadas por GLUT. Este toolkit también provee mecanismos para interacción del usuario con Gazebo vía dispositivos estándar de entrada como teclados o ratones.

El Mundo

Un ambiente completo es la colección de modelos y sensores. El terreno y las construcciones representan modelos estacionarios; los robots y otros objetos son dinámicos; los sensores se separan de la simulación dinámica ya que sólo colectan o emiten información. A continuación, se hace una descripción de los componentes en el simulador:

- *Modelo*: es cualquier objeto que tiene una representación física, desde objetos con simple geometría hasta robots complejos. Los modelos están compuestos de por lo menos un cuerpo rígido, cero o más juntas, sensores y de una interfaz para facilitar el flujo de datos.

- *Cuerpo*: representa una construcción básica de bloques de un modelo. Su representación física toma formas entre cubos, esferas, cilindros, planos y líneas. Cada cuerpo tiene una masa asignada, fricción, factor de rebote y propiedades como color, textura, transparencia, etc.
- *Juntas*: son las que proveen al mecanismo conexión entre cuerpos para formar relaciones cinemáticas y dinámicas. Hay múltiples juntas tales como prismáticas, rotacionales, de bolas, entre otras. Las juntas deben ser seleccionadas adecuadamente ya que a la vez pueden generar inestabilidad al conectar varias, haciendo que se pierda estabilidad si se seleccionan los parámetros inadecuados.
- *Interfaz*: permite que el usuario pueda acceder y controlar los modelos. Los comandos enviados por la interfaz pueden hacer que el modelo mueva las juntas, cambie la configuración de los sensores en él, o pedir información de los sensores.
- *Sensores*: En Gazebo no tienen representación física, pero sin un sensor un robot no podría realizar ninguna tarea útil, por lo que su importancia en el modelo es esencial y mandatoria. Los principales sensores en Gazebo son:
 - ✓ Odómetro: da información de la distancia recorrida.
 - ✓ Rayo de proximidad: devuelve el punto de contacto del objeto más cercano a lo largo de la trayectoria del rayo.
 - ✓ Cámara: esta representa una escena usando OpenGL desde la perspectiva del modelo en la que está.

Construcción de Modelos

Los modelos son creados seleccionando los cuerpos y juntas necesarios para garantizar funcionalidad, de tal manera que se pueda evaluar y probar el robot en escenarios peligrosos y difíciles. Para crear el modelo no basta únicamente con unir enlaces, también es necesario conocer los controladores del robot, para que estos puedan ser ejecutados y llamados desde ROS o Matlab como un *.launch*. En este mismo ámbito de construcción de modelos también se deben tener en cuenta los sensores que son parte fundamental del modelo en construcción. En la figura 4

se muestra un modelo que permite ser manipulado y seguir trayectorias preestablecidas; creado a partir de código XML y con estructura xacro, con el que es posible captar imágenes del entorno, para luego ser procesadas.

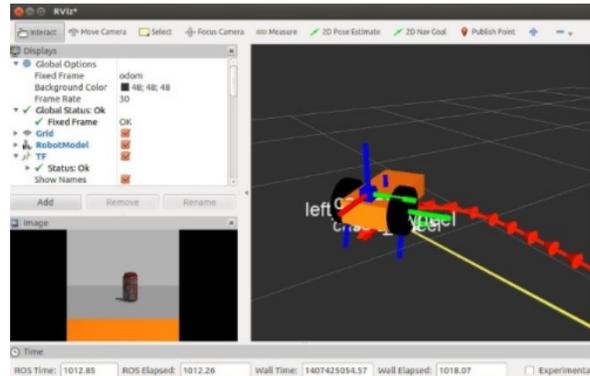


Figura 4 Modelo creado con sensores y sistema.

Integración Matlab/Windows con Gazebo

El principal aspecto a considerar para llevar a cabo la integración de Matlab con Gazebo es que se ejecuta Matlab desde un sistema operativo Windows y Gazebo desde Linux, usando para tal propósito una máquina virtual (MV) como VMware Player. En la figura 5 se muestra el panorama general de la infraestructura empleada para llevar a cabo la integración. Dentro del sistema cliente Linux, se encuentra instalado Kubuntu, incluyendo ROS (Hydro) y Gazebo; donde la conexión se establece a través de la IP que tiene asignada la imagen ejecutada en VMware Player. Con esto, es posible enviar mensajes ROS desde Windows (Matlab), hasta Gazebo.

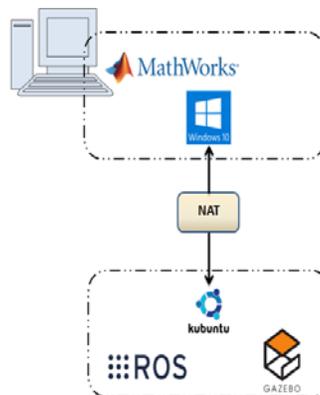


Figura 5 Infraestructura para conectar Gazebo con Matlab en Windows a través de una máquina virtual con Linux.

Simulación en Gazebo

Al ejecutar Gazebo en el entorno Linux, se puede hacer la conexión con la IP desde Matlab:

```
rosinit('IP_Maquina_virtual')
```

Cuando se logra la conexión descrita anteriormente, se pueden enviar los mensajes al entorno de Gazebo para poder simular el ambiente y las tareas deseadas. Para la cual primero es necesario hacer correr el nodo Gazebo desde Linux. Una vez hecho esto ya se puede tener información del entorno virtual como los topics que hay disponibles para suscribirse y poder publicar o bien recibir información de ellos como sería el caso de la cámara. Desde Matlab esto se logra a partir de la creación de objetos que permiten la fácil interacción y control con el simulador. Para lo anterior se cuenta con los siguientes métodos:

- Servicio para pausar la simulación.
- Servicio que permite retomar la simulación después de haber sido pausada.
- Servicio para resetear la simulación.
- Servicio para reiniciar el mundo.
- Método empleado para retornar la estructura de las propiedades físicas de la simulación.
- Asignar propiedades físicas a la simulación.
- Lista de los elementos creados.
- Método para eliminar un modelo específico.

Del mismo modo, se cuenta con otras propiedades de gran utilidad:

- Parámetros físicos de la simulación
- Lista de los modelos del mundo Gazebo.
- Bandera para comprobar si los servicios están ejecutándose.
- Servicio cliente para aplicar fuerzas.
- Servicio cliente para aplicar torques a las juntas.
- Servicio cliente para establecer el estado del modelo.
- Servicio cliente para establecer las propiedades del modelo.

- Servicio cliente para configurar las juntas.

3. Resultados

Mediante Gazebo y Matlab, se simuló el robot Jackal (ver figura 6), que es un robot que cuenta con un computador, GRS e IMU; es pequeño y tiene una plataforma de búsqueda en terreno pesado. El robot ha sido simulado con la finalidad de recorrer campos de cultivo y hacer un procedimiento de reconocimiento de características fenológicas de plantas.



Figura 6 Robot Jackal.

Al robot Jackal simulado en Gazebo se le agregó una Point Grey Flea3, la cual es una cámara equipada con un sensor de 640x480 píxeles que permite la toma de 120 imágenes por segundo. Es pequeña y liviana, lo que permite la fácil adaptación sobre el robot. En la figura 7 se muestra el resultado de instalar la cámara mencionada en la plataforma robótica Jackal.

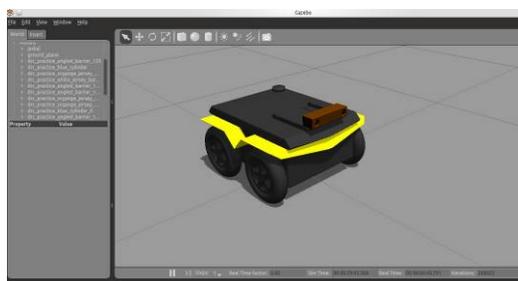


Figura 7 Robot Jackal en Gazebo con Point Grey Flea3.

El ambiente que se recreó corresponde a un cultivo; entorno por el cual el robot Jackal debía explorar, adquiriendo fotografías de cada una de las plantas que se encuentran en la zona, para de esta forma detectar deficiencias ya sea nutricionales o causadas por plagas y enfermedades, figura 8.

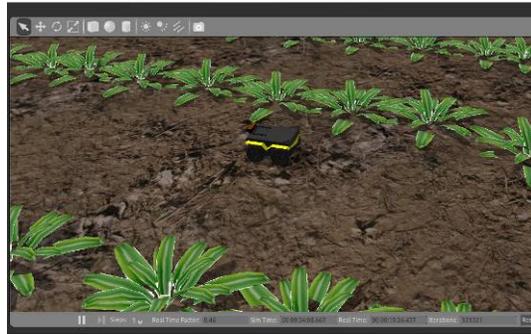
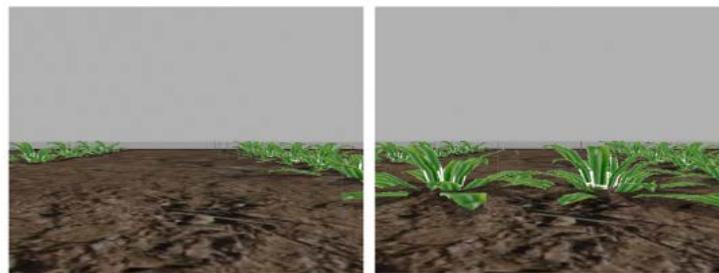


Figura 8 Robot Jackal explorando entorno de cultivo.

Para la adquisición de las imágenes se procede a realizar la conexión con Matlab. Conociendo la dirección IP de la Máquina Virtual desde donde está siendo ejecutado el nodo de Gazebo, se puede hacer un llamado de todos los topics, suscribir la cámara y obtener las imágenes requeridas. Para este caso el topic es: `/front/lef t/image_raw`. Las imágenes obtenidas se muestran en la figura 9.



(a) Adquisición lejana a las plantas (b) Adquisición realizada desde un ángulo medio



(c) Adquisición en una posición cercana a la planta

Figura 9 Imágenes, cámara Bumblebee 2 desde robot Jackal simulado en Gazebo.

La manera en la que el robot Jackal logra recorrer todo el campo del cultivo es suscribiéndose a su topic IMU, un sensor propio del Jackal que permite obtener

información acerca de la velocidad, orientación y fuerzas gravitacionales. Al obtener datos desde este sensor, se puede publicar a los topics `/jackal_velocity_controller/cmd_vel` y `/jackal_velocity_controller/odom` manteniendo una correcta ruta para completar su recorrido de manera autónoma.

A medida que el Jackal adquiere las fotografías, estas pueden ser procesadas para dar alerta inmediata de aquellas que no se encuentran en un estado apropiado. Para esto se emplea el procesador de imágenes de Matlab, el cual define por medio de espacios de color, si la planta está en buen estado o no. En la figura 10 se aprecia un procesamiento básico para la determinación de la región correspondiente a la vegetación basado en segmentación por banda espectral verde.

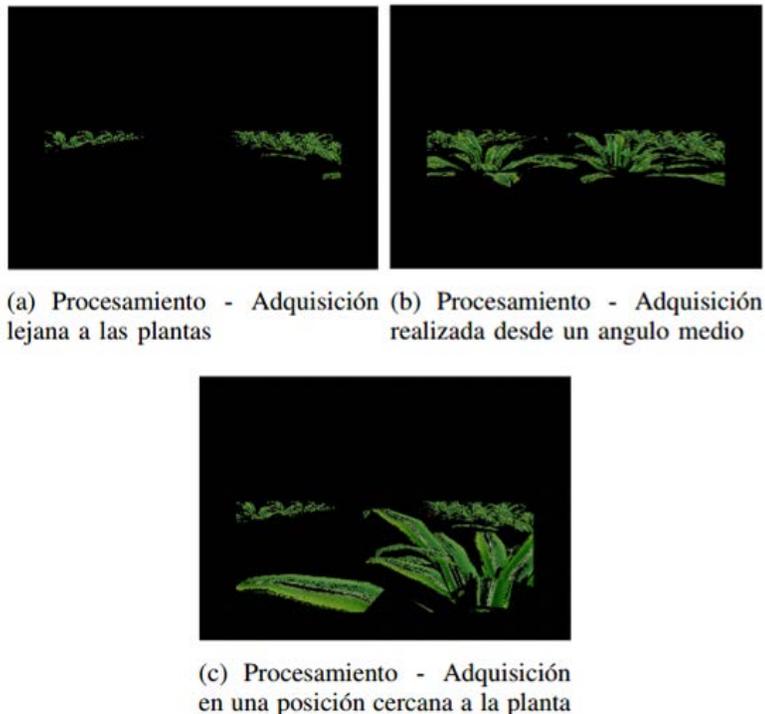


Figura 10 Imágenes procesadas mediante matlab para caracterización de las plantas.

En la figura 11 se aprecia la secuencia de detección de planta con complicaciones fenológicas. En este caso corresponde a un efecto de extremo daño del comportamiento. El algoritmo del sistema de reconocimiento está desarrollado para ser calibrado a condiciones estándar de color para plantas sanas. Al detectar la región correspondiente a vegetación, el algoritmo permite la definición de grado

de severidad del daño al comparar áreas ocupadas por plantas de cultivo (secuencia intermedia de la figura 11, con respecto al área total de vegetación sana (secuencia inferior de la figura 11). Se establece por lo tanto que, como en el ejemplo de la figura 11, existe mayor área de vegetación, con respecto a la vegetación sana, por lo que la planta se considera que presenta alguna dificultad.



Figura 11 Secuencia de determinación de daño presente en una planta del cultivo.

4. Discusión

Partiendo de un escenario robótico que permitiera llevar a cabo la agricultura de precisión se logra desarrollar el control del robot Jackal, haciendo que este recorra los terrenos indicados, reconociendo a su vez aspectos asociados a la vegetación para luego poder analizarlos y permitir que este llegue a una toma de decisiones inteligentes.

El robot seleccionado es el adecuado para su tarea debido su tamaño y costo. Este mismo puede ser llevado a un mismo ambiente, pero con robots cooperativos. Así como lo menciona Yiang Tan, el progreso de las tecnologías hardware y esquemas cooperativos facilita la interacción entre robots, así como la interacción con el ambiente que los rodea [Tan, 2015].

Para poder lograr la localización del robot en el entorno se realiza un algoritmo similar al que hace Weihua Yang: a partir de la odometría del robot o por medio del reconocimiento de elementos conocidos, se tiene un estimado de la ubicación, que luego es procesada por medio de los sensores con los que cuenta el robot Jackal permitiendo obtener una posición y una orientación [Yang, 2008] que puede ser manipulada para la finalidad de este trabajo.

Existen diferentes algoritmos de automando, por lo que el más adecuado es aquel que permita el mando del robot teniendo en cuenta que las condiciones del ambiente no son siempre las mismas, así que lo más apropiado es parametrizar aquellas características del ambiente que pueden ser constantes. Pero en general se trata de crear un sistema para el robot que sea capaz de hacer frente a situaciones cambiantes del entorno [Rodríguez, 2004]

Con esta información se logra que el robot realice tres tareas fundamentales:

- Estimar su posición y orientación
- Mantener actualizado el mapa del entorno
- Detectar los posibles obstáculos

Para soportar y garantizar las tareas fundamentales mencionadas, se hace uso fundamental de Matlab para el procesamiento de imágenes y para la comunicación con el simulador que también es tema principal de este trabajo.

En cuanto a la comunicación entre Matlab y el entorno creado en Gazebo/ROS que es totalmente exitosa, vale la pena mencionar que es necesario verificar las variables de ROS MASTER y ROS MASTER URI, las cuales al no estar bien configurada puede traer inconvenientes para tener acceso a todos los topics disponibles. Además de mantener la imagen de la máquina virtual siempre actualizada y estable, con esto se lleva a cabo una simulación exitosa que permite obtener los resultados mostrados en este trabajo y continuar investigando en ellos. Se pretende seguir trabajando en el sistema de navegación del robot, mejorando su algoritmo de estimación de posición y orientación, además del reconocimiento de imágenes para poder identificar más acertadamente las plantas deseadas entre las no deseadas.

Agradecimientos

Se agradece a la Dirección de Investigación y Extensión de la Universidad Nacional de Colombia sede Bogotá por la financiación del proyecto institucional titulado: Modelamiento Basado en Agentes para Aplicaciones de Agricultura de Precisión, de la Convocatoria Nacional de Proyectos para el Fortalecimiento de la Investigación, Creación e Innovación de la Universidad Nacional de Colombia

2016-2018. Cárdenas P. F. expresa su agradecimiento a Colciencias por la beca de estudio de doctorado en el extranjero 2007. Jiménez A. F. expresa su agradecimiento a la gobernación de Boyacá, por la beca de doctorado de la convocatoria para la formación de capital humano de alto nivel para el departamento de Boyacá 2015 y también a la Universidad de los Llanos.

5. Conclusiones

La integración Matlab - Gazebo es posible gracias a la comunicación por NAT entre los ambientes involucrados, además de contar con la herramienta Robotics de Matlab, que permite contar con todas las funcionalidades de ROS y así asociarlas a un entorno de simulación.

Jackal es un robot pequeño con una amplia capacidad de exploración, lo que permite aprovechar dicha ventaja en campos de difícil acceso como terrenos rocosos. Lo que lo hace un robot adecuado para tareas como las que se mencionan en este artículo

Al hacer que Gazebo funcione desde Matlab, permite realizar procesamientos de información e imágenes, aprovechando la potencialidad y los toolbox de esta herramienta de software. Gazebo no se limita sólo a que el estudiante utilice los modelos existentes por defecto, sino que también permite la posibilidad de crear modelos personalizados.

Para este procedimiento es importante comprender los procedimientos y la organización de la información de las carpetas propias de esta herramienta.

Se logró desarrollar el control para un robot que recorre terrenos en espacio abierto, además de permitir mediante la misma simulación el reconocimiento de aspectos asociados a la vegetación, que al ser procesadas y analizadas permiten desarrollar herramientas de soporte a decisiones inteligentes.

6. Bibliografía y Referencias

- [1] H., J. J., & S. S., Teaching Robot Kinematic in a Virtual Environment. San Francisco : Proceedings of the World Congress on Engineering and Computer Science 2010 Vol I, 2010.

- [2] F. R., & M. B., Control y robótica en agricultura. Almeria: Universidad de Almeria, servicio de publicaciones, 2005.
- [3] K., Robot Operating System (ROS), Cham, Suiza: Springer International Publishing, 2016.
- [4] J. R., Blog de programación y robótica, Obtenido de El Simulador Stage de ROS: <https://jjromeromarras.wordpress.com/2014/09/14/el-simulador-stage-de-ros/>, 14 de septiembre de 2014.
- [5] M. E., Evolutionary Humanoid Robotics, Berlin, Springer, 2015.
- [6] N. K., & A. H., Design and use paradigms for Gazebo, an open-source multi-robot simulator, Intelligent Robots and Systems, 2004, Proceedings, 2004 IEEE/RSJ International Conference on. Sendai, Japón: IEEE, 2004.
- [7] R. V., 2015, Wiki: stage: <http://wiki.ros.org/stage>.
- [8] V. M., 2015, Robotic simulation scenarios with Gazebo and ROS: <http://www.generationrobots.com/blog/en/2015/02/robotic-simulation-scenarios-with-gazebo-and-ros/>.
- [9] W. Y., Autonomous Robots Research Advances. Nueva York: Nova Science Publisher, Inc, 2008.
- [10] Y. T., Handbook of Research on Design, Control, and Modeling of Swarm Robotics, Peking, China: Information Science Reference, 2015.