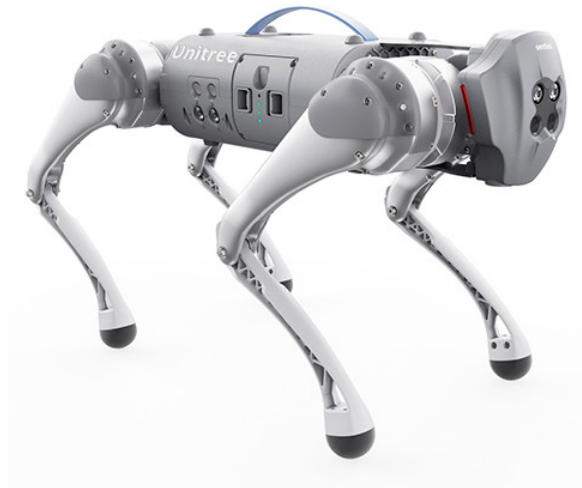


Trabajo teórico/práctico Robots Móviles.

Robots cuadrupedos

Curso 2021/2022 / Ingeniería robótica



1.Introducción	1
2. Diseño mecánico y análisis de parámetros	2
A. Diseño mecánico.	2
B. Análisis cinemático y dinámico.	3
3. Cálculo de trayectorias	5
A. Trayectoria del centroide	5
B. Cálculo de la trayectoria de las 4 patas	7
C. Algoritmos de control	7
4. Implementación en la máquina virtual	8
A. Uso de Chvmp_setup_assistant	9
B. Uso del paquete con la demo de chvmp	10
C. Empleo de nuestro propio paquete	11

Santiago Pérez Gisbert

Mario Jerez Tallón

José García Jávega

1.Introducción

En la actualidad, el diseño de sistemas de locomoción de robots caminantes se ha derivado del estudio de sistemas biológicos, especialmente de animales terrestres comunes, en los cuales se puede observar un sistema de locomoción basado en un conjunto de eslabones y articulaciones denominadas patas.

La naturaleza ha encontrado a través de millones de años de evolución un sistema óptimo para el desplazamiento sobre terrenos irregulares, basado en el apoyo discreto de las patas sobre el terreno. Las máquinas caminantes aprovechan este tipo de locomoción ya sea mediante uno o más puntos de apoyo (monópodos, bípedos, cuadrúpedos, hexápodos, etc.), convirtiéndolo las máquinas ideales para espacios peligrosos y de difícil acceso.

Por otra parte, estas máquinas exigen sistemas mecánicos y de control más complejos que los usados por sistemas con ruedas, así como mayores niveles de sensorización, que hasta el momento limitan que sean industrialmente rentables, ya que los costos de fabricación y mantenimiento son mucho más elevados que otros tipos de robots.

Las principales características son:

- Gran omnidireccionalidad.
- Capacidad de adaptación al terreno.
- Control de las fuerzas en los puntos de apoyo.
- Apoyos discretos en el terreno.

La mayoría de robots caminantes están destinados a la exploración de terrenos irregulares inaccesibles para el ser humano. Entre las principales aplicaciones de estas máquinas están la detección de minas personales, transportación de equipos y herramientas de construcción, trabajos de forestación, limpieza de fachadas, entre otros. Por todas las características mencionadas las máquinas caminantes se convierten en mecanismos idóneos para desplazarse no solo por superficies horizontales, sino también por superficies con cierto grado de inclinación o completamente verticales. En este último caso, estos mecanismos toman el nombre de robots escaladores.

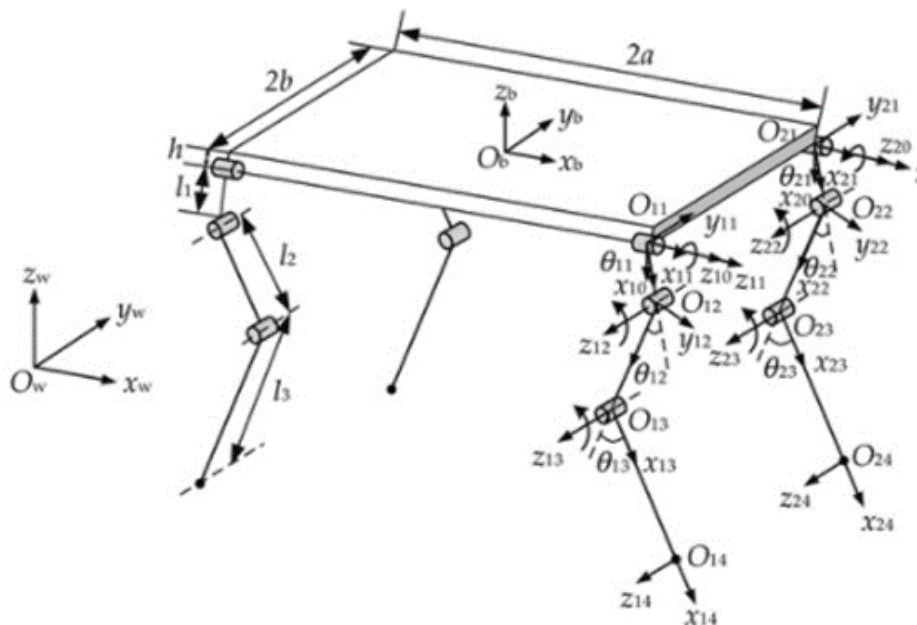
A continuación, se expone una breve descripción del diseño e implementación de la plataforma móvil como herramienta de investigación en modos de caminar de robots cuadrúpedos, y que permita además el estudio de métodos de locomoción para robots escaladores.

2. Diseño mecánico y análisis de parámetros

A. Diseño mecánico.

Un mayor número de patas, permite mayores velocidades y mejor estabilidad a los robots caminantes, sin embargo, incrementa su complejidad mecánica y de control. Un robot cuadrúpedo disminuye la complejidad mecánica del sistema frente a robots caminantes con mayor número de patas, manteniendo las características de omnidireccionalidad, adaptación al terreno y estabilidad estática y dinámica. La estructura básica está constituida por el cuerpo y cuatro patas. Cada pata está conformada por cuatro eslabones, tres de los cuales se encuentran unidos a través de articulaciones rotacionales operadas por el movimiento del eje de un motor, dando un total de 12 grados de libertad a todo el sistema. El último eslabón (eslabón 4) está unido a su predecesor por medio de una articulación universal no actuada que proporciona dos grados de libertad más al extremo final de la pata. Una plataforma cuadrada permite la instalación de las cuatro patas y de los dispositivos electrónicos requeridos para el control, constituyéndose en la base estructural del cuerpo del robot.

Las patas están dispuestas en el cuerpo de forma simétrica en cada esquina de manera que el centro de gravedad se encuentre en el eje central del cuerpo y que los esfuerzos realizados por cada pata se distribuyan uniformemente. Se ha diseñado el sistema para optimizar el consumo energético, mediante el uso de motores con mecanismos autobloqueantes, lo que permite que no exista consumo de energía cuando el robot se encuentra detenido.



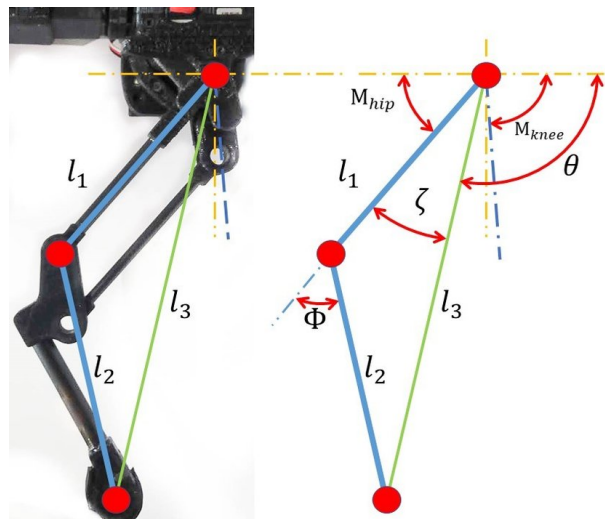
A continuación, se detallan las características más importantes consideradas para el diseño del robot:

- Bajo consumo de energía.
- Gran omnidireccionalidad sobre terrenos horizontales.
- Flexibilidad en el diseño mecánico de manera que la modificación de cualquier eslabón se realice sin dificultades.
- Flexibilidad en el diseño electrónico que permite generar plataformas estándar de control.
- Patas ligeras y capaces de soportar cargas elevadas en distintas posiciones.
- Bajo costo
- Capacidad de transportar una carga

El movimiento generado por las articulaciones se ve limitado por la estructura mecánica de los eslabones. La amplitud máxima alcanzada por cada una de las articulaciones, de igual forma, los ángulos máximos y mínimos generados por las articulaciones nos permiten determinar la altura máxima y mínima del cuerpo respecto a una superficie horizontal, y la altura de paso del robot. Esta última nos permite determinar la altura máxima de los objetos que el robot podrá superar en un paso normal. La amplitud de los ángulos conseguidos en cada una de las articulaciones, permite una diversidad de configuraciones que puede adoptar el robot al momento de la locomoción.

B. Análisis cinemático y dinámico.

En la siguiente imagen se presenta el esquema cinemático de la pata, el cual nos permite identificar las ecuaciones de la cadena cinemática tanto en coordenadas cartesianas como en coordenadas articulares, considerando únicamente aspectos geométricos. El análisis dinámico por su parte, permite determinar los esfuerzos, así como velocidades y aceleraciones que se generan en las diferentes articulaciones debido a los esfuerzos realizados, sean estos debido a los momentos de inercia, centros de gravedad, fuerzas externas, entre otras.



Para obtener la cinemática inversa de las patas del robot se emplean las siguientes ecuaciones:

$$\begin{aligned}\theta_1 &= \text{atan}^{-1} \frac{a_z}{a_x} \\ \theta_2 &= \text{sen}^{-1} \left[\frac{l_3 \text{sen } \theta_3}{\sqrt{\left(\sqrt{a_x^2 + a_z^2} - l_1\right)^2 + a_y^2}} \right] \\ &\quad + \text{sen}^{-1} \left[\frac{a_y}{\sqrt{\left(\sqrt{a_x^2 + a_z^2} - l_1\right)^2 + a_y^2}} \right] \\ \theta_3 &= \cos^{-1} \left[\frac{\left(\sqrt{a_x^2 + a_z^2} - l_1\right)^2 + a_y^2 - l_2^2 - l_3^2}{2l_2l_3} \right]\end{aligned}$$

Donde:

- **l1, l2, l3 y l4** son las longitudes de los cuatro eslabones respectivos que conforman la pata.
- **θ1, θ2, y θ3** son los desplazamientos angulares de las articulaciones uno, dos y tres respectivamente.
- **ax, ay y az** son las coordenadas de la posición del extremo final de la pata a calcular
- Con estas ecuaciones se pueden desarrollar trayectorias para así poder mover el robot

También podemos obtener la matriz jacobiana del siguiente modo:

$$\mathbf{J} = \begin{bmatrix} 0 & l_2 c_2 + l_3 c_{23} & l_3 c_{23} \\ c_1(l_1 + l_2 c_2 + l_3 c_{23}) & -s_1(l_2 s_2 + l_3 s_{23}) & -l_3 s_1 s_{23} \\ -s_1(l_1 + l_2 c_2 + l_3 c_{23}) & s_1(l_2 s_2 + l_3 s_{23}) & l_3 c_1 s_{23} \end{bmatrix}$$

Esta matriz se puede utilizar para obtener el vector de velocidades siguiendo la siguiente fórmula

$${}^bV_i = {}^b\dot{p}_i = J(\theta_i)\dot{\theta}_i$$

Donde \mathbf{p} representa la velocidad del pie del robot.

También es posible calcular la velocidad angular con la siguiente fórmula

$$\dot{\theta}_i = J^{-1}(\theta_i){}^bV = J^{-1}(\theta_i){}^b\dot{p}_i$$

Donde la inversa de la matriz jacobiana se obtendrá de la siguiente forma

$$J^{-1} = \begin{bmatrix} 0 & c_1/(l_1 + l_2c_2 + l_3c_{23}) & s_1/(l_1 + l_2c_2 + l_3c_{23}) \\ s_{23}/(l_2s_3) & s_1c_{23}/(l_2s_3) & -c_1c_{23}/(l_2s_3) \\ -(l_2s_2 + l_3c_{23})/(l_2l_3s_3) & -s_1(l_2c_2 + l_3c_{23})/(l_2l_3s_3) & c_1(l_2c_2 + l_3c_{23})/(l_2l_3s_3) \end{bmatrix}$$

3. Cálculo de trayectorias

A. Trayectoria del centroide

La primera trayectoria que se deberá de tener en cuenta será la trayectoria del centro de masas, situado en el centro del cuerpo del robot, ya que este presenta una gran influencia en la estabilidad en los robots cuadrúpedos, para calcular correctamente su trayectoria se deberá calcular tanto su posición horizontal como vertical, para la obtención de la posición horizontal se emplea la siguiente fórmula:

$$\begin{cases} x_{bw}(t) = d_0 + d_1t + d_2t^2 + d_3t^3 & 0 \leq t \leq T_m \\ x_{bp}(t) = c_0 + c_1t + c_2t^2 + c_3t^3 & T_m \leq t \leq T \end{cases}$$

Donde T es el tiempo de un ciclo de movimiento y T_m es la fase de oscilación. Para poder hacer este cálculo se deberá tener en cuenta que se trata de un movimiento continuo, dando por hecho que la velocidad y el desplazamiento horizontal del centro de masas son equivalentes

Para el cálculo de la posición vertical se emplea la siguiente fórmula

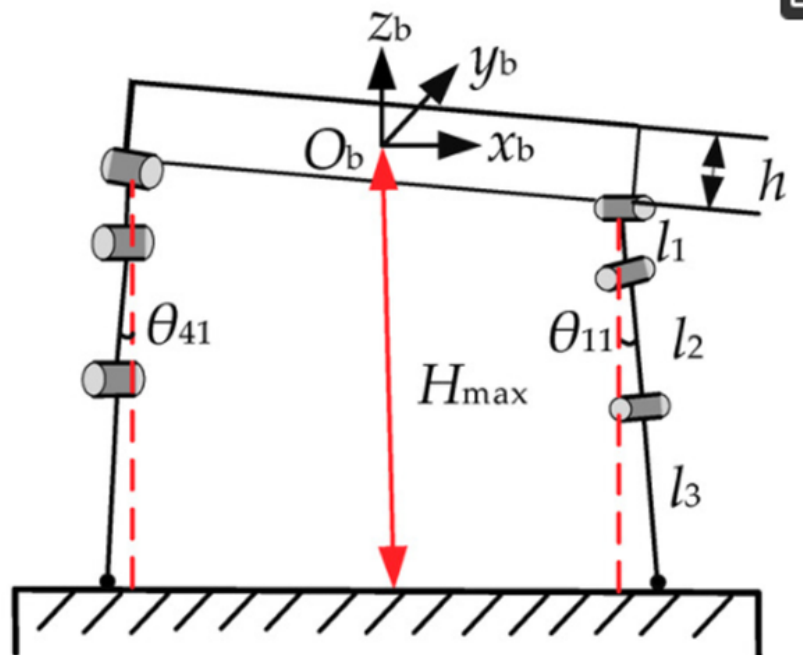
$$\begin{cases} z_{bw}(t) = H_{bz} & 0 \leq t \leq T_m \\ z_{bp}(t) = H_{bz} & T_m \leq t \leq T \end{cases}$$

Donde H_{bz} es la altura del robot a usar, se puede determinar una altura máxima con el fin de evitar singularidades con la fórmula que se destaca a continuación:

$$H_{\max} = \frac{1}{2} [h \cos q_2 + (l_1 + l_2 + l_3) (\cos \theta_{11} + \cos \theta_{41})]$$

Donde q_2 es el ángulo del tronco de nuestro robot

Aquí se puede apreciar mejor la altura descrita anteriormente:



De manera similar, debido a la simetría de la estructura del robot cuadrúpedo hidráulico, mientras que el robot está sostenido por cuatro patas, la altura del centroide del tronco es la mitad de la altura total de las dos patas apoyadas en diagonal.

B. Cálculo de la trayectoria de las 4 patas

Del mismo modo, se puede calcular la trayectoria de las cuatro patas, cada una de estas cuenta con 3 grados de libertad, y teniendo en cuenta la cinemática analizada se pueden calcular las trayectorias de las articulaciones a través de la trayectoria de los “pies” del robot.

Dicho esto, hay varios métodos para calcular las trayectorias de los pies, pero en este caso se destacará el siguiente:

$$\begin{cases} p_x = S \left(\frac{t}{T_m} - \frac{1}{2\pi} \sin(2\pi \frac{t}{T_m}) \right) - \frac{S}{2} & 0 \leq t \leq T_m \\ p_y = 0 & 0 \leq t \leq T_m \\ p_z = 2H \left[\frac{t}{T_m} - \frac{1}{4\pi} \sin(4\pi \frac{t}{T_m}) \right] & 0 \leq t \leq \frac{T_m}{2} \\ p_z = 2H \left[1 - \frac{t}{T_m} + \frac{1}{4\pi} \sin(4\pi \frac{t}{T_m}) \right] & \frac{T_m}{2} \leq t \leq T_m \end{cases}$$

Donde solo se deberá destacar que H es la altura máxima que alcanza el pie del robot.

C. Algoritmos de control

A la hora de controlar el robot cuadrúpedo, se buscará tratar de controlar su centroide, se supone que durante la marcha el ángulo pitch y yaw del tronco son nulos, su control es más fácil cuando están equipados con un conjunto de elementos virtuales como un resorte y una amortiguación, gracias a estos elementos se puede ignorar la no linealidad de las patas, y para hacerlo más simple se ignorará la inercia de las patas.

Por tanto, la relación entre el momento de torsión articular y la fuerza generalizada que actúa sobre el tronco puede expresarse como:

$$\boldsymbol{\tau} = -\mathbf{J}^T \mathbf{F}$$

De donde podemos obtener la fórmula para obtener el momento de la pata que se encuentra apoyada del siguiente modo:

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -c_1 & -c_1 \\ 0 & s_1 & s_1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

Combinando las fórmulas anteriores se puede obtener que

$$\begin{bmatrix} F \\ T \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 0 & -s_{23}/(l_2 s_3) & (l_2 s_2 + l_3 s_{23})/(l_2 l_3 s_3) \\ -c_1/(l_1 + l_2 c_2 + l_3 c_{23}) & -s_1 c_{23}/(l_2 s_3) & s_1(l_2 c_2 + l_3 c_{23})/(l_2 l_3 s_3) \\ -s_1/(l_1 + l_2 c_2 + l_3 c_{23}) & c_1 c_{23}/(l_2 s_3) & -c_1(l_2 c_2 + l_3 c_{23})/(l_2 l_3 s_3) \\ -1 & 0 & 0 \\ 0 & -c_1 & -c_1 \\ 0 & s_1 & s_1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

4. Implementación en la máquina virtual

Para lograr mover un robot cuadrúpedo se comenzó trabajando con el robot de 4 patas unitree aliengo, sin embargo, este carecía de un controlador para su simulación, por lo que se trató de crear un pequeño controlador que calculase la cinemática inversa para así obtener la posición de sus patas y poder mover el robot por un entorno vacío indicando posiciones articulares a cada una de sus patas, sin embargo debido a que no se tenían en cuenta factores como el rozamiento con el suelo y demás errores que pudiesen ocurrir a lo largo de la simulación se acabó descartando este método, ya que requería de un controlador PID y aun así almacenaremos mucho error.

Es por ello, que optamos por un paquete denominado CHVMP con el cual se puede controlar cualquier tipo de robot con cuatro patas, sin embargo, para ello se debía de instalar en un primer caso este paquete correctamente además de otro paquete denominado chvmp_setup_assistant, el cual nos ayudará con la creación de nuestro propio paquete empleando el archivo urdf de cualquier robot que queramos implementar.

Para la creación de este paquete propio simplemente tendremos que facilitar nuestro fichero urdf además de la altura máxima a la que se puede mover el tronco de nuestro robot.

Una vez instalado el paquete inicial, podemos lanzar en la interfaz de Rviz una simulación de los controladores, sin embargo, destacaremos el empleo de Gazebo para mapear empleando SLAM.

En nuestro entorno, que en este caso se tratará de un mapa similar a una obra en gazebo como se ve a continuación:



Dentro de este entorno, podremos mover con total libertad el robot que deseemos, mapeando la zona y observando que detecta el radar si lo tiene nuestro robot, el método para movernos puede ser o mediante teclado de forma similar a como lo hacíamos con el turtlebot o especificando en la interfaz de rviz que se abre de forma paralela con la función 2DNavGoal, con la que el robot detectará la trayectoria a seguir para alcanzar el punto con la orientación que le hemos especificado.

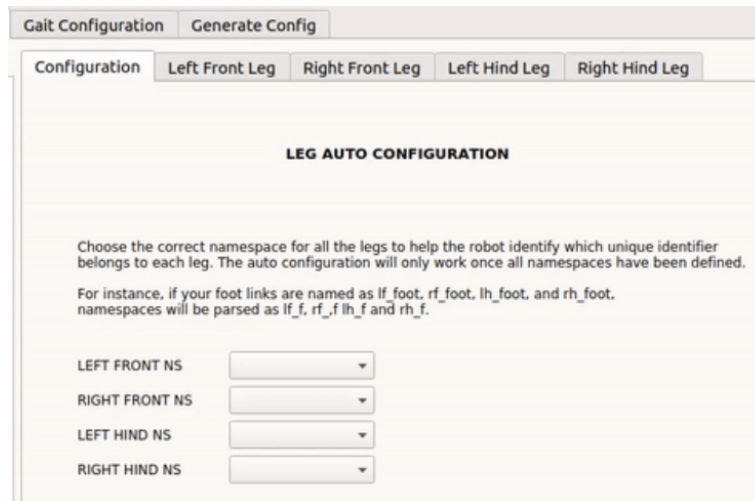
A. Uso de Chvmp_setup_assistant

Para crear nuestro propio paquete con el robot cuadrupedo que deseemos se emplea este paquete, el cual se comporta de un modo similar al asistente de moveit, en este indicaremos el archivo urdf del robot a utilizar y el programa se encargará de generar los distintos controladores de este.

Para lanzarlo ejecutamos en terminal el siguiente comando:

```
roslaunch champ_setup_assistant setup_assistant.launch
```

Una vez lanzado, se especificará la ruta del urdf, el tamaño de las articulaciones y algunos otros detalles de estas como su orientación. A continuación, se especifica el namespace de cada pata para que estas se configuren automáticamente, una vez hecho esto el paquete se encargará de crear un paquete con el nombre que le indiquemos con el que podremos ejecutar los mismos comandos que en la demo de chvmp especificada previamente. Interfaz donde se indican los distintos namespace:



B. Uso del paquete con la demo de chvmp

Empleando este paquete, se moverá un robot cuadrupedo por el entorno ya mencionado en gazebo, para mover el robot por el entorno activaremos la interfaz de Rviz para indicar la posición que queremos alcanzar, tras esta orden el robot calculará el camino más corto esquivando los obstáculos, sin embargo, al indicar un camino excesivamente largo en la máquina virtual puede ir algo lento, aun así alcanza la posición indicada sin ningún problema.

Para ejecutar la demo seguiremos los siguientes pasos:

`roslaunch champ_config gazebo.launch`

Con este comando, cargaremos el entorno de gazebo con el robot contenido en el paquete y cargaremos sus controladores, a continuación tendremos que lanzar en otra terminal uno de los siguientes comandos, dependiendo de si queremos mapear el entorno o si simplemente queremos navegar por el:

`roslaunch champ_config slam.launch rviz:=true`

Con este comando lanzaremos SLAM, empleando la interfaz de Rviz controlaremos con facilidad la posición a la que queremos que el robot vaya.

`roslaunch champ_config navigate.launch rviz:=true`

Con este comando lanzaremos la navegación, empleando la interfaz de Rviz controlaremos con facilidad la posición a la que queremos que el robot vaya.

Un ejemplo de esta ejecución es la siguiente:

- <https://drive.google.com/file/d/18uOLSRPKPpOpvggktdsXTLsfXa5uFab-/view?usp=sharing>

En el cual podemos observar que están detectando los distintos sensores del robot.

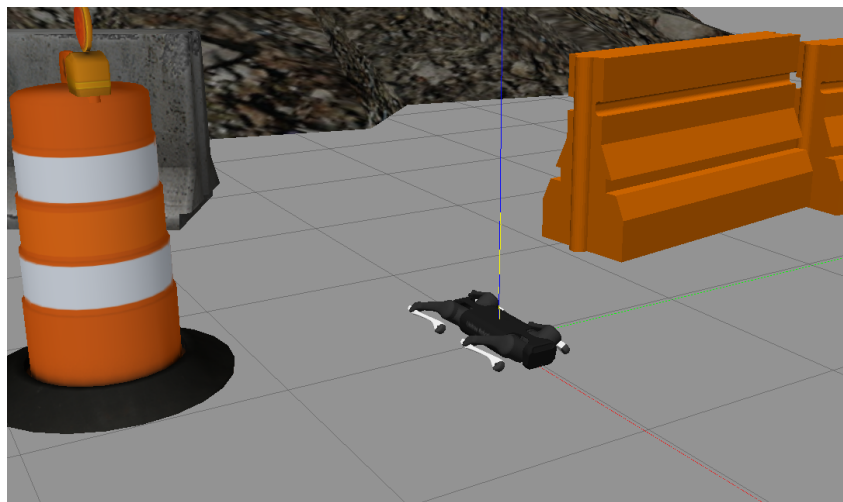
Por otra parte, si empleamos el paquete en lugar de para navegar lo empleamos con el fin de mapear, no observaremos el total del mapa dentro de la interfaz de Rviz, sino que solo podremos apreciar las áreas que los sensores van escaneando.

C. Empleo de nuestro propio paquete para mover el robot Aliengo de la empresa unitree robotics

Finalmente, se creó nuestro propio paquete con el robot de unitree, llamado aliengo, en este se presentaron algunos problemas a la hora de cargar los controladores en el entorno de gazebo, donde uno de los errores se soluciona modificando el archivo gazebo.xacro de el mismo, sin embargo, tras estas modificaciones no fue posible hacer que el robot camina indicando las posiciones, por ello decidimos comprobar si los controladores funcionan correctamente moviendo el robot por un entorno vacío donde las órdenes serán enviadas por teclado como se puede apreciar a continuación:

- <https://drive.google.com/file/d/1DQKpnsuRmCi13kGCTWmwoBsmc4TLMmr-/view?usp=sharing>

Como se puede apreciar, no es problema de los controladores, ya que estos se han creado con éxito, sin embargo no hemos logrado emplearlos en el entorno de gazebo, quedando el robot en la posición inicial con las patas sin ejercer ninguna fuerza como se puede ver en esta imagen:



Algunos robots cuádrupedos a implementar son los siguientes, destacados en los enlaces donde se puede descargar el urdf que puede ser empleado para simular la tarea mencionada.

- GoogleAI ROBEL D'Kitty
- MIT Mini Cheetah
- OpenDog V2
- Open Quadruped
- SpotMicroAI
- Stochlab's Stochlite
- MangDang's Mini Pupper
- Anybotics' ANYmal B
- Anybotics' ANYmal C
- Boston Dynamic's Little Dog
- Boston Dynamic's Spot
- Dream Walker