

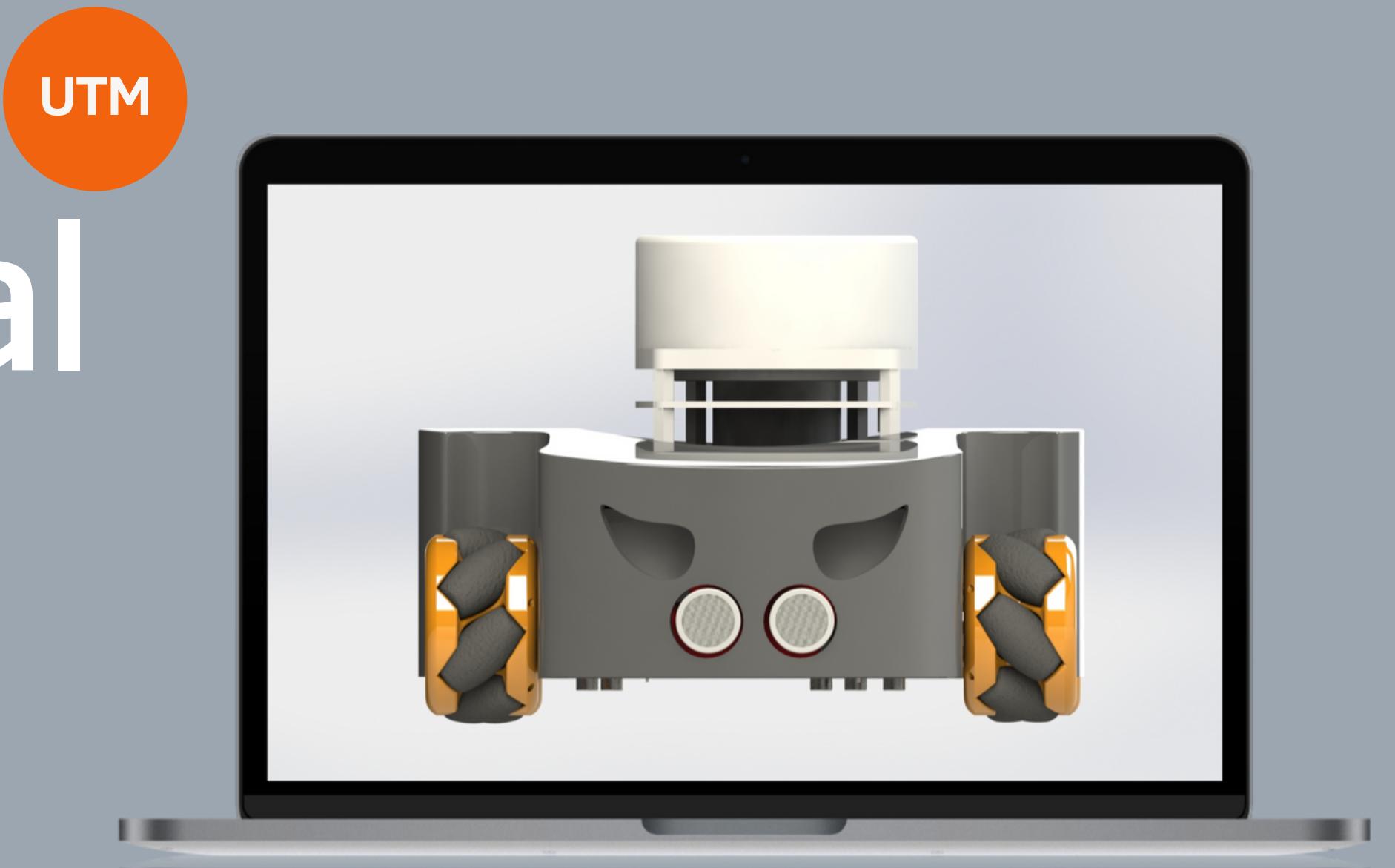


Robot móvil omnidireccional "Astérix"

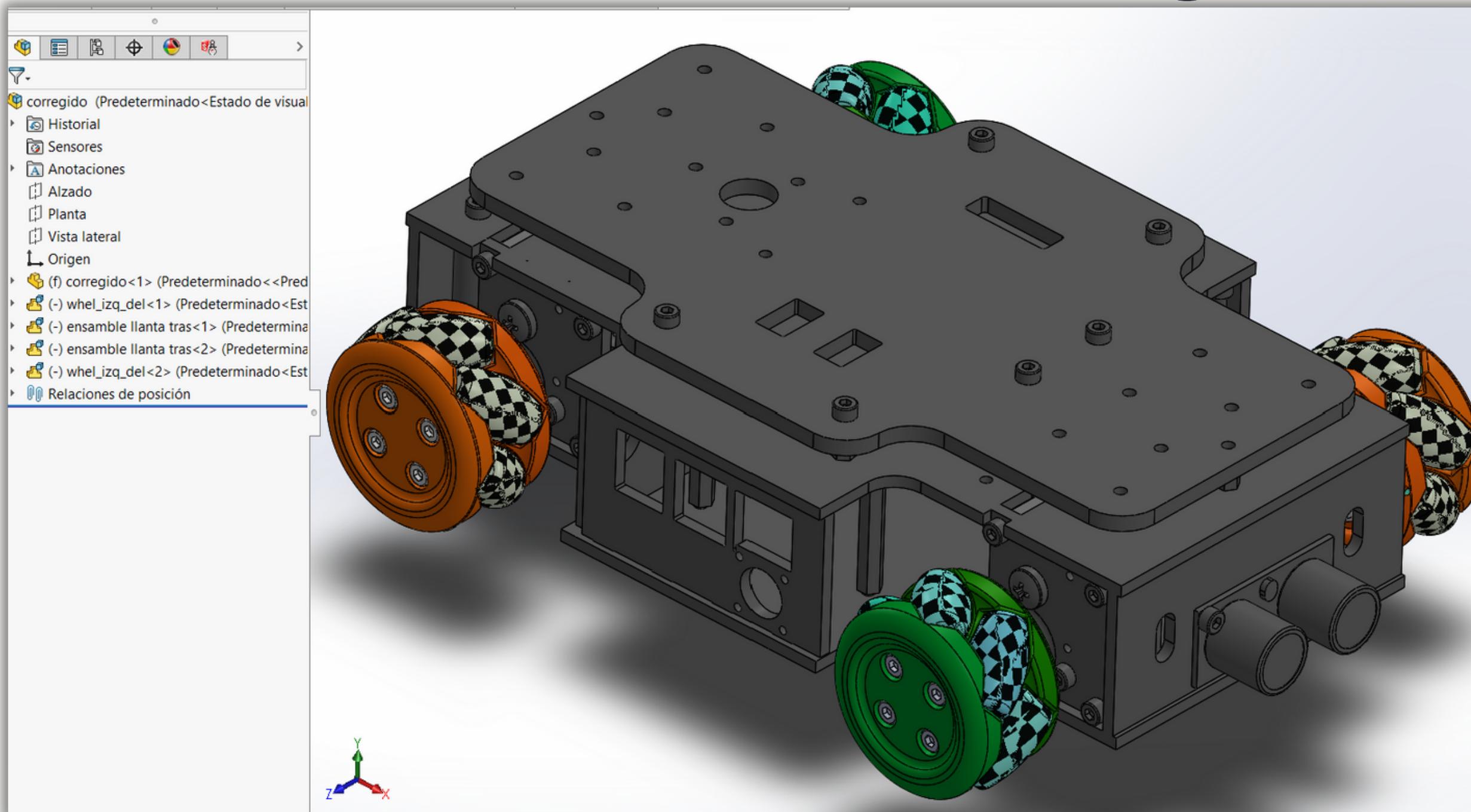
Cruz Juárez David Ricardo
Dionicio Pizarro Jorge Alberto
Hernández López Oliver Edson

ESTUDIANTES DE INGENIERÍA MECATRÓNICA.
DÉCIMO SEMESTRE

01.



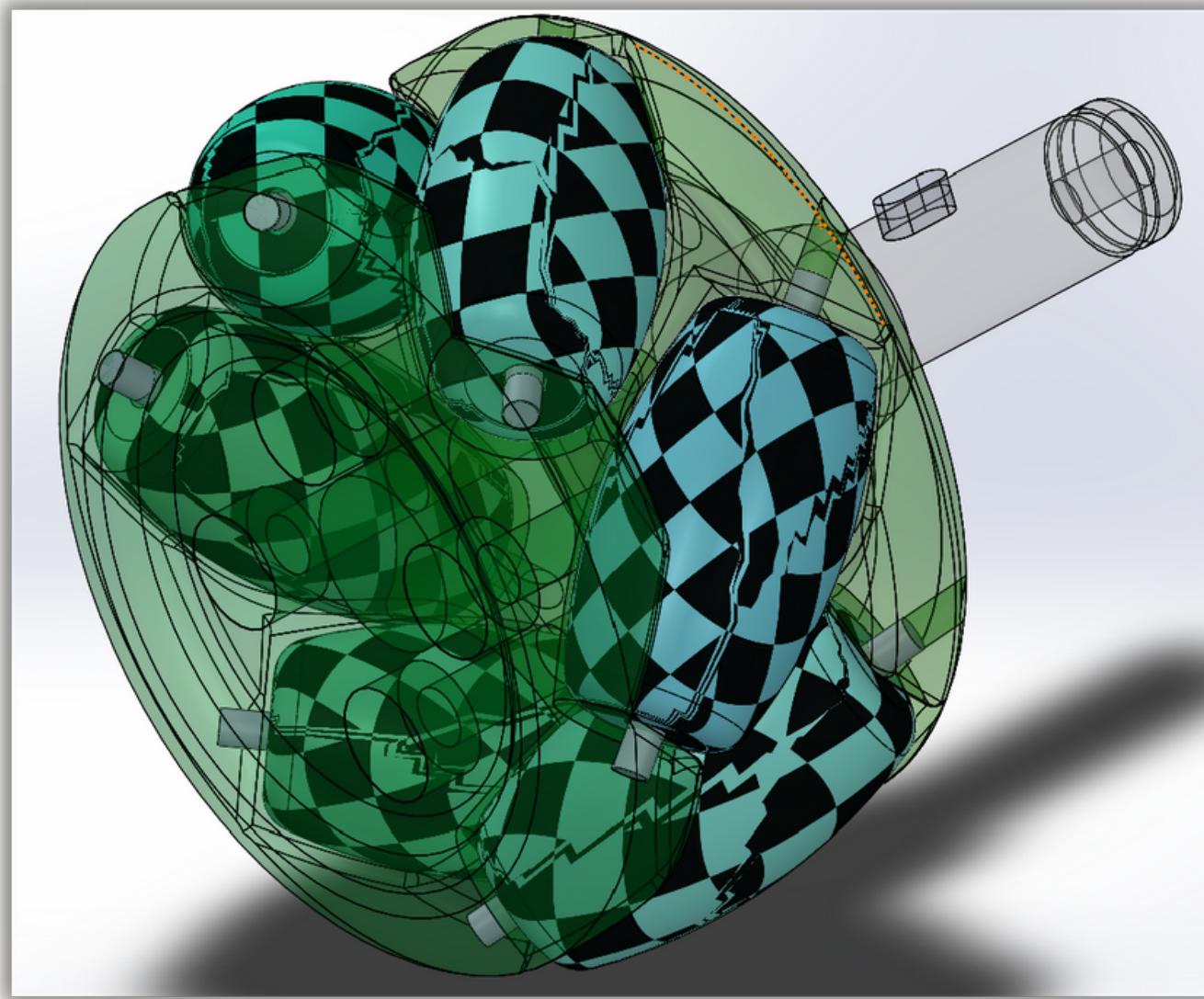
Posicionamiento de origen



02.

ROBOT OMNIDIRECCIONAL "ASTÉRIX" JUNIO DE 2021

Subensamble rueda mecanum

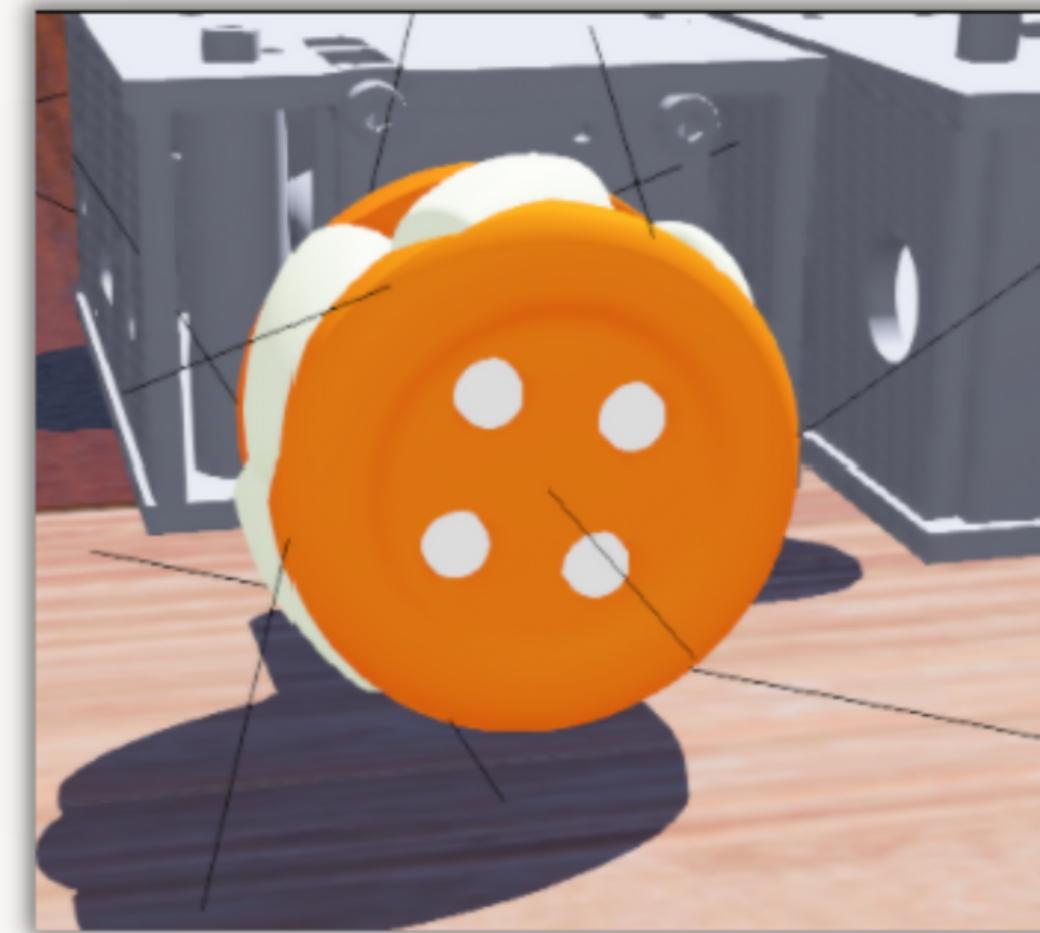
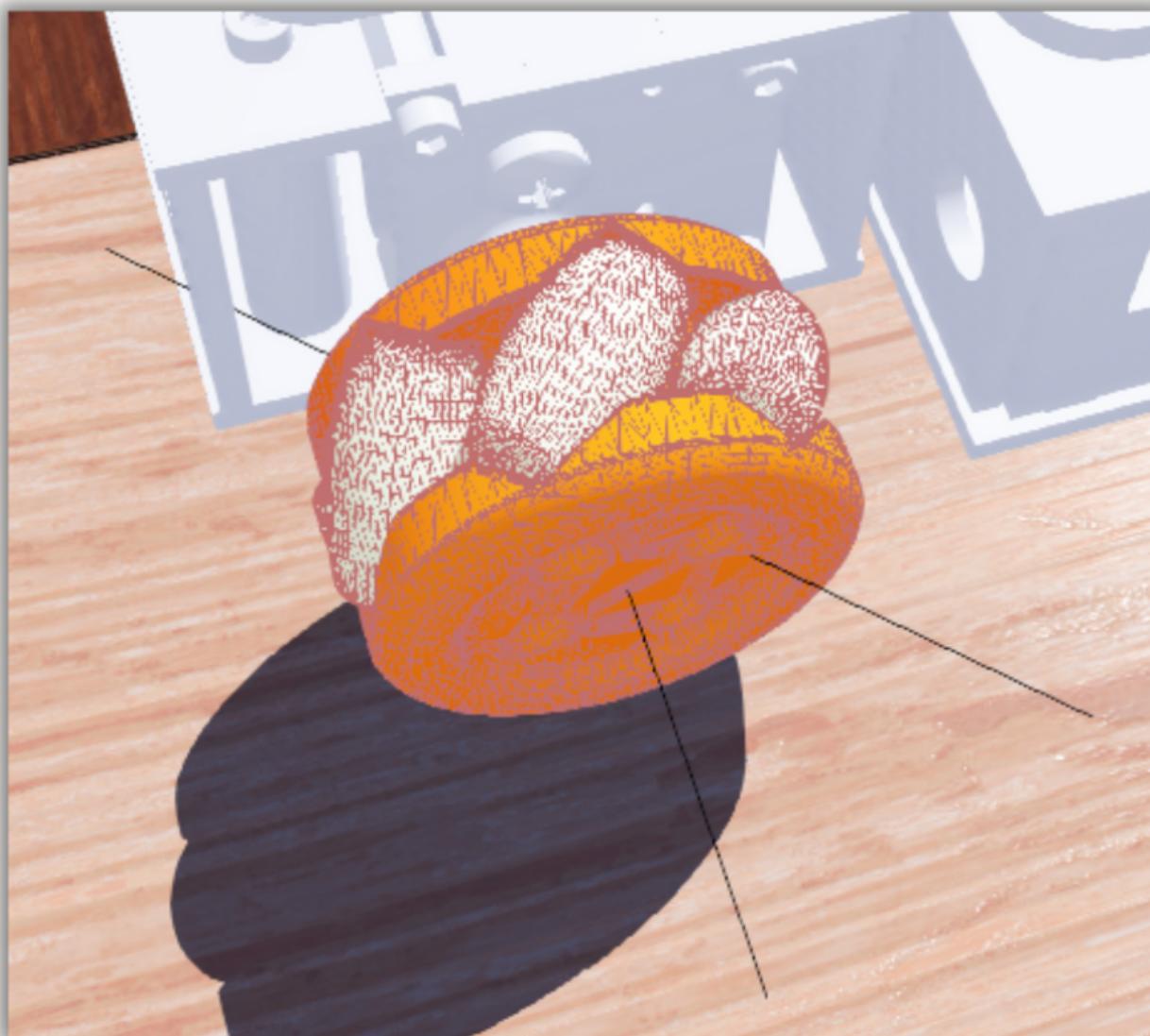


03.

ROBOT OMNIDIRECCIONAL "ASTÉRIX" JUNIO DE 2021

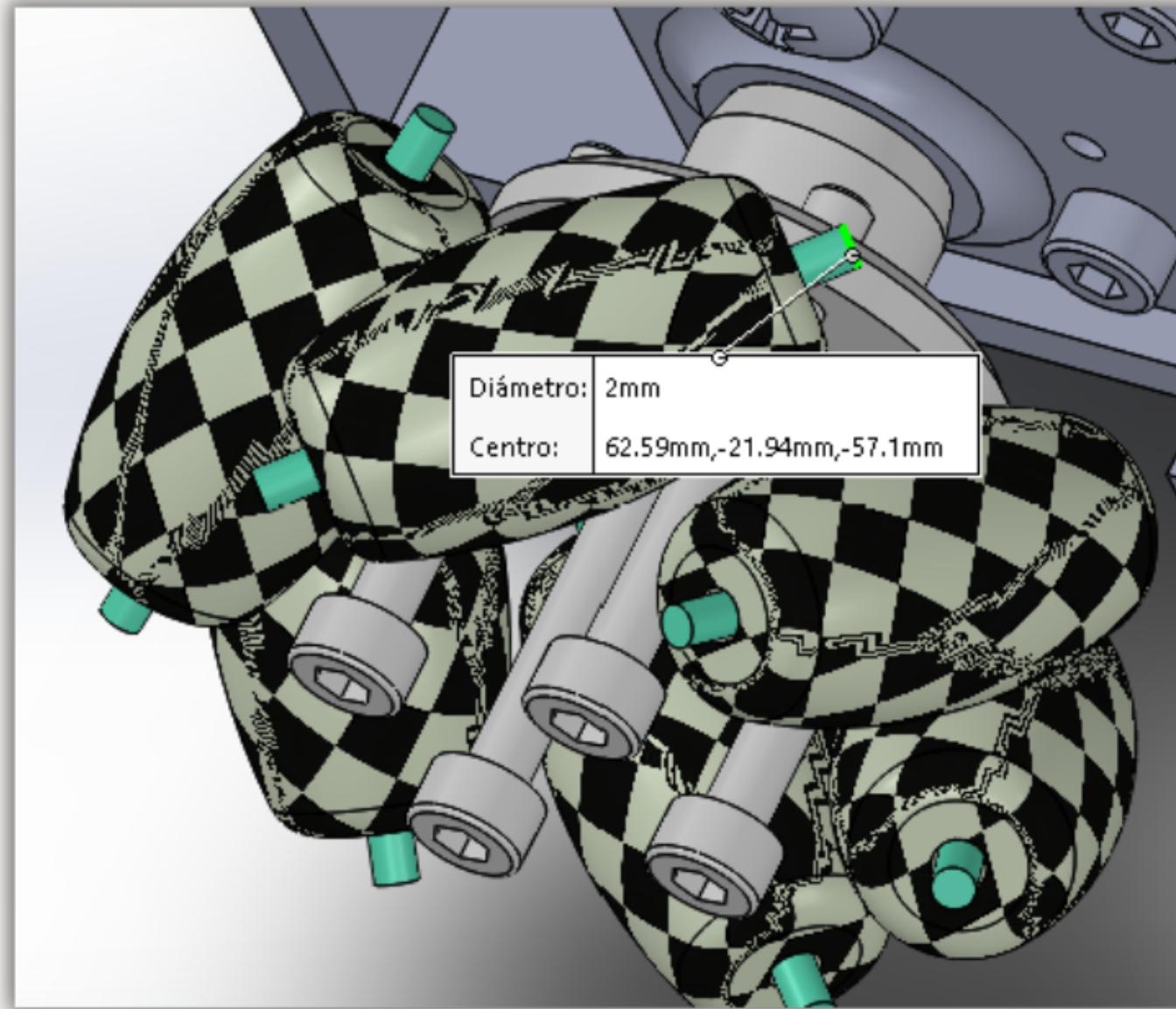
Paradigma de trabajo

Para nuestro análisis utilizamos dos formas de movimiento relativo: asignando ejes y uniones independientes a cada rodillo de la rueda mecanum (forma convencional) y el uso de un eje virtual en dirección de la velocidad resultante (utilizado por CoppeliaSim).



Configuración de los ejes de acción rodillo por rodillo (forma convencional)

≡



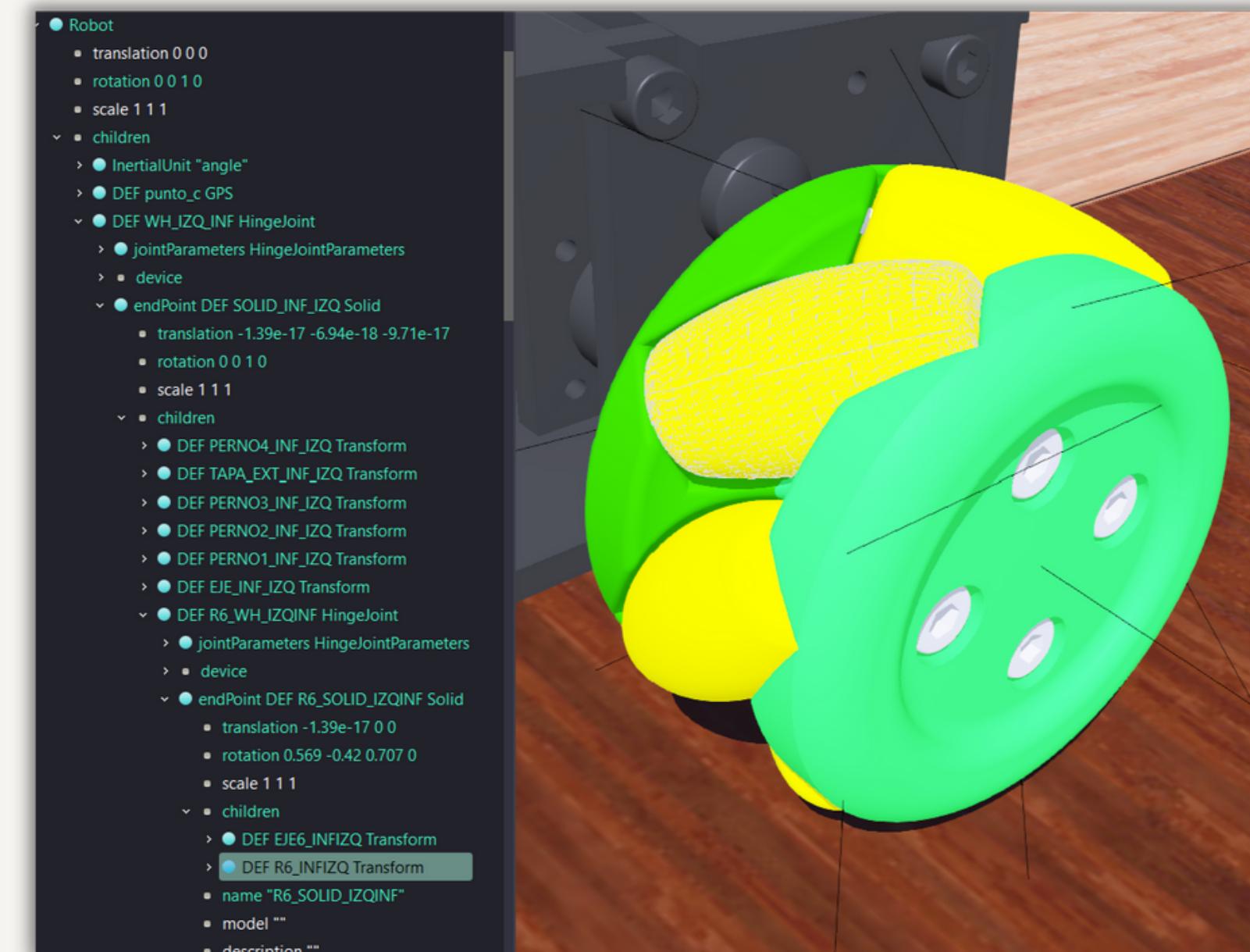


```
%% Inicio del programa  
  
C1 = v1-v2;  
  
C1 = C1/norm(C1);  
  
C2 = ([ (v1(1)+v2(1))/2 (v1(2)+v2(2))/2 (v1(3)+v2(3))/2]);  
C1 =  
.  
-0.5657 -0.4243 -0.7071  
  
>> C2  
  
C2 =  
  
14.0000 21.5000 32.5000
```

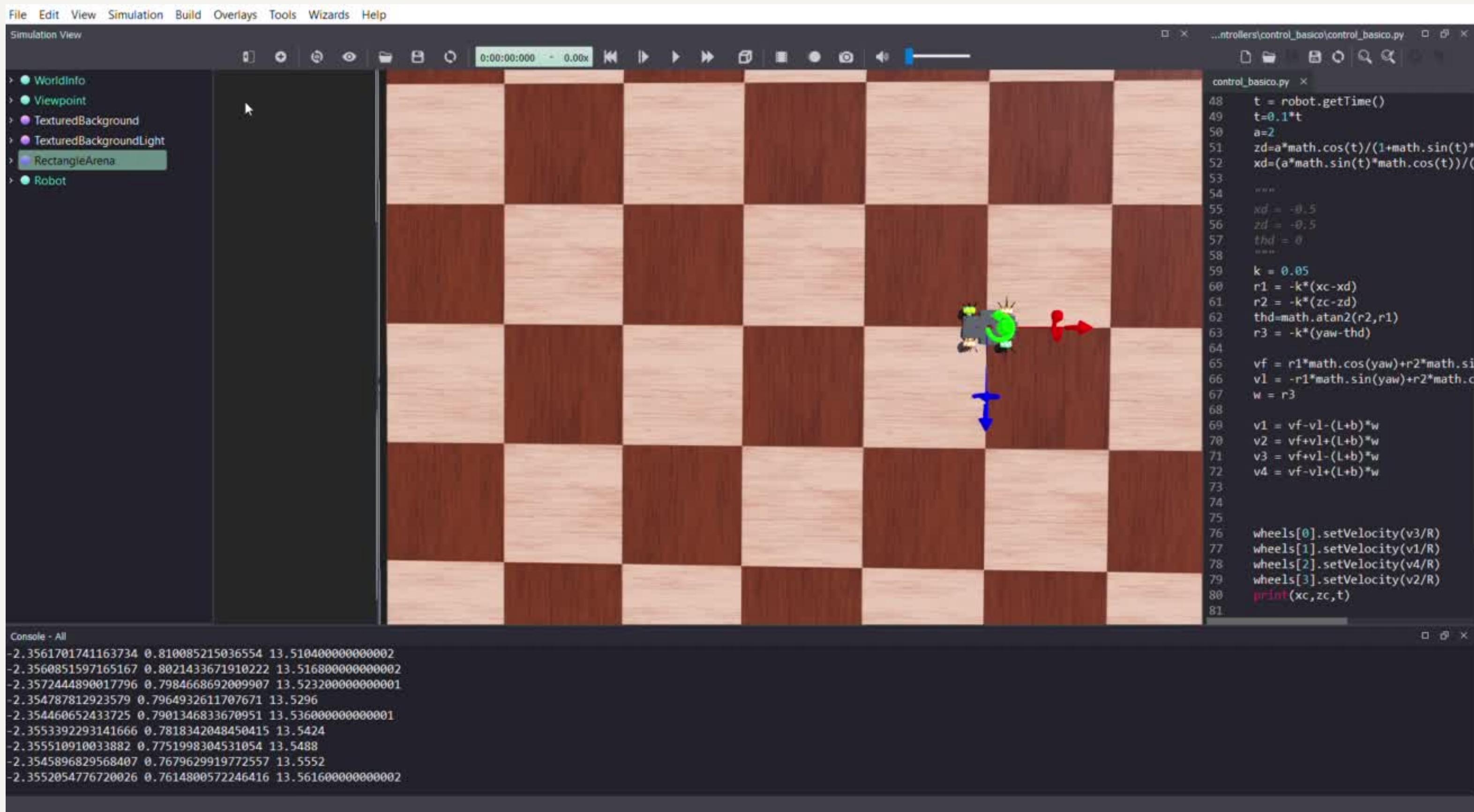
Debemos asegurarnos que cada eje esta perfectamente centrado, para ello se realizó un pequeño programa en Matlab, donde se obtiene la distancia el vector normalizado del eje de acción del rodillo y el punto central del eje del rodillo (Parametros *Anchor* y *Axis*).

Caracterización de interacciones

Se colocó como objeto colisionable los ejes y rodillos. Esto asegura que la unión entre ellos no se salga de su posición. Además de que permite a los rodillos interactuar (colisionar) con el entorno, permitiendo el movimiento del robot

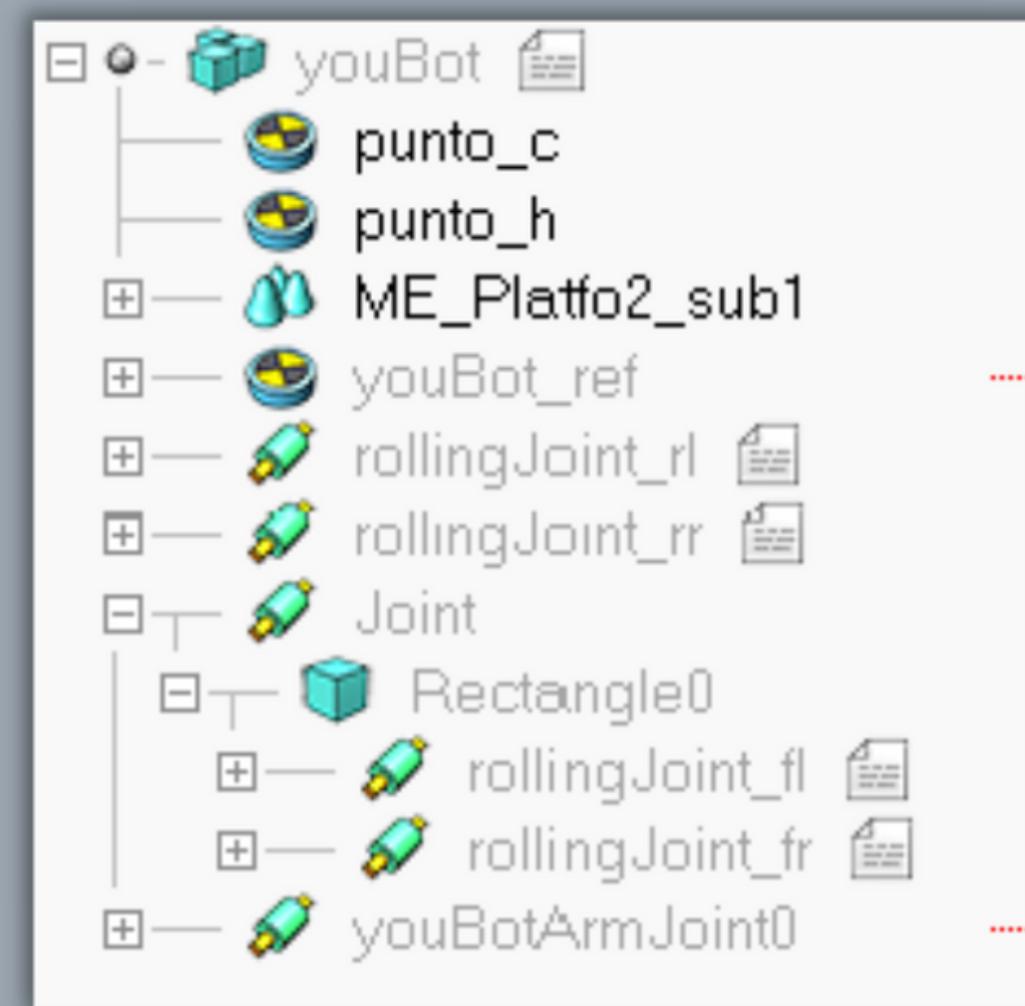


Video de los resultados



Configuración de eje central de la rueda a 45°(usado por CoppeliaSim)

La propuesta hecha por el modelo de ejemplo YouBot en CoppeliaSim



Cada una de las juntas presenta una estructura de 4 partes fundamentales:



```

function sysCall_actuation()
    sim.resetDynamicObject(wheel)

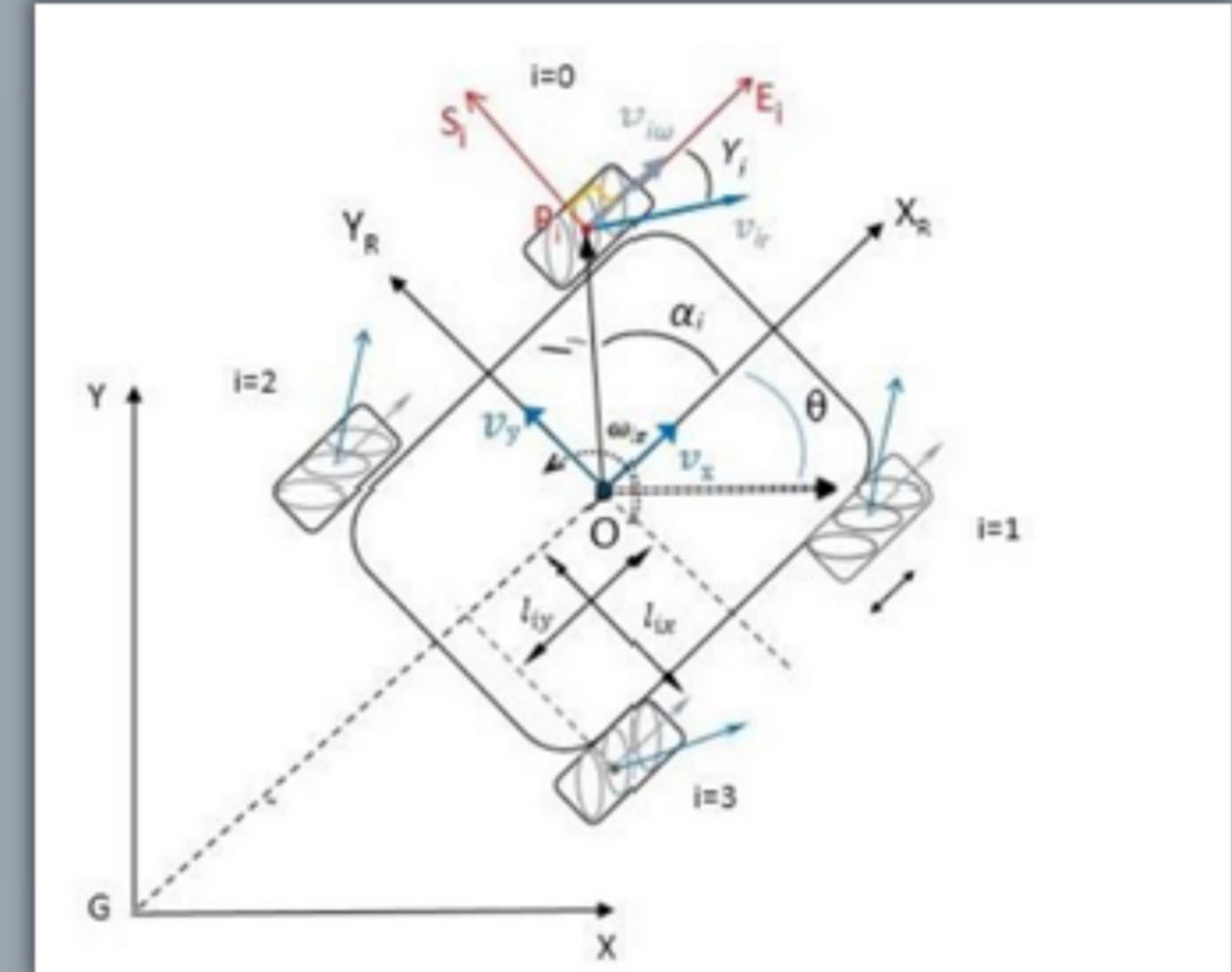
    sim.setObjectPosition(slipping+sim.handleflag_reljointbaseframe,rolling,{0,0,0})
    sim.setObjectOrientation(slipping+sim.handleflag_reljointbaseframe,rolling,{math.pi/4,0,math.pi})

    sim.setObjectPosition(wheel+sim.handleflag_reljointbaseframe,rolling,{0,0,0})
    sim.setObjectOrientation(wheel+sim.handleflag_reljointbaseframe,rolling,{0,0,0})
end

```

El eje del motor principal de la rueda tiene orientación cero y el eje virtual (el eje de acción del conjunto de rodillos) tiene una orientación de 45 grados

Configuración programada en LUA para cada



Una vez colocados los nodos de manera jerárquica es necesario configurar la dirección de los ejes de acción para ambos movimientos, para ello es necesario modificar dos parámetros: Axis y Anchor.



jointParameters HingeJointParameters

- position 0
- axis 0 0 1
- anchor -0.0756 -0.0388 0.0676

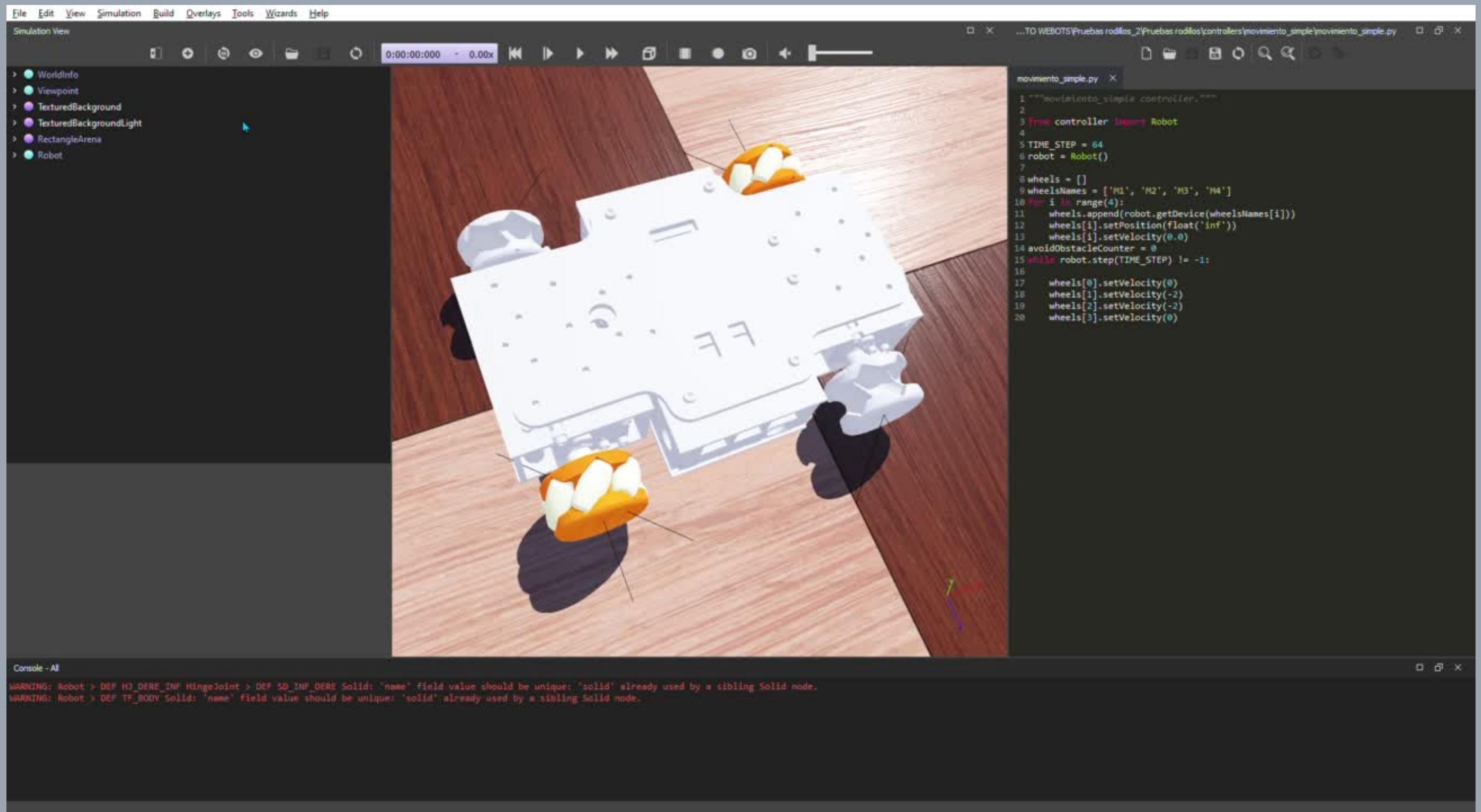
jointParameters HingeJointParameters

- position 0
- axis 0.707 0 0.707
- anchor -0.0756 -0.0388 0.0676

Es claro ver que la orientación del eje de 45 grados es bastante similar a la especificada en CoppeliaSim. Por tanto, no debería haber complicación alguna

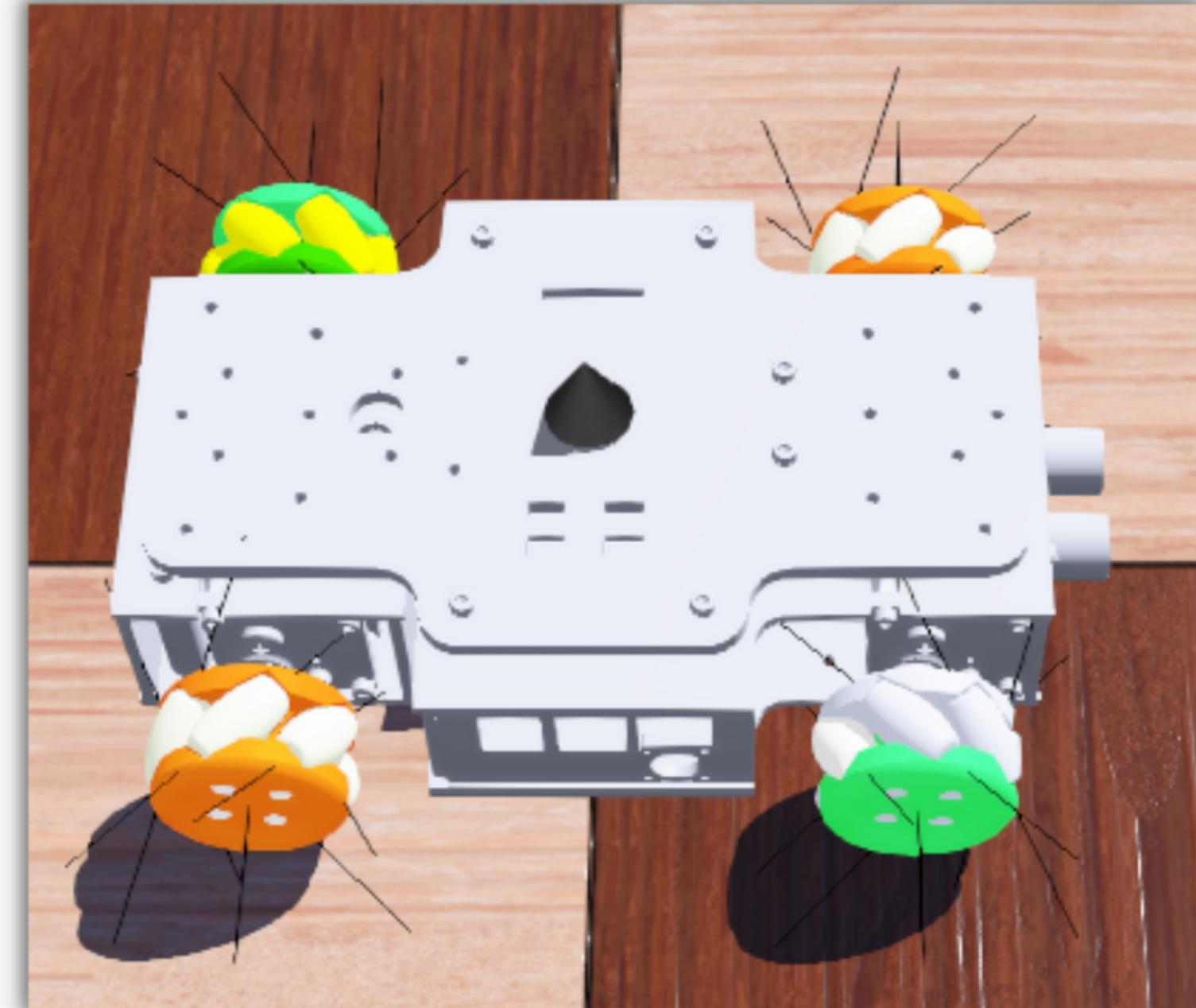
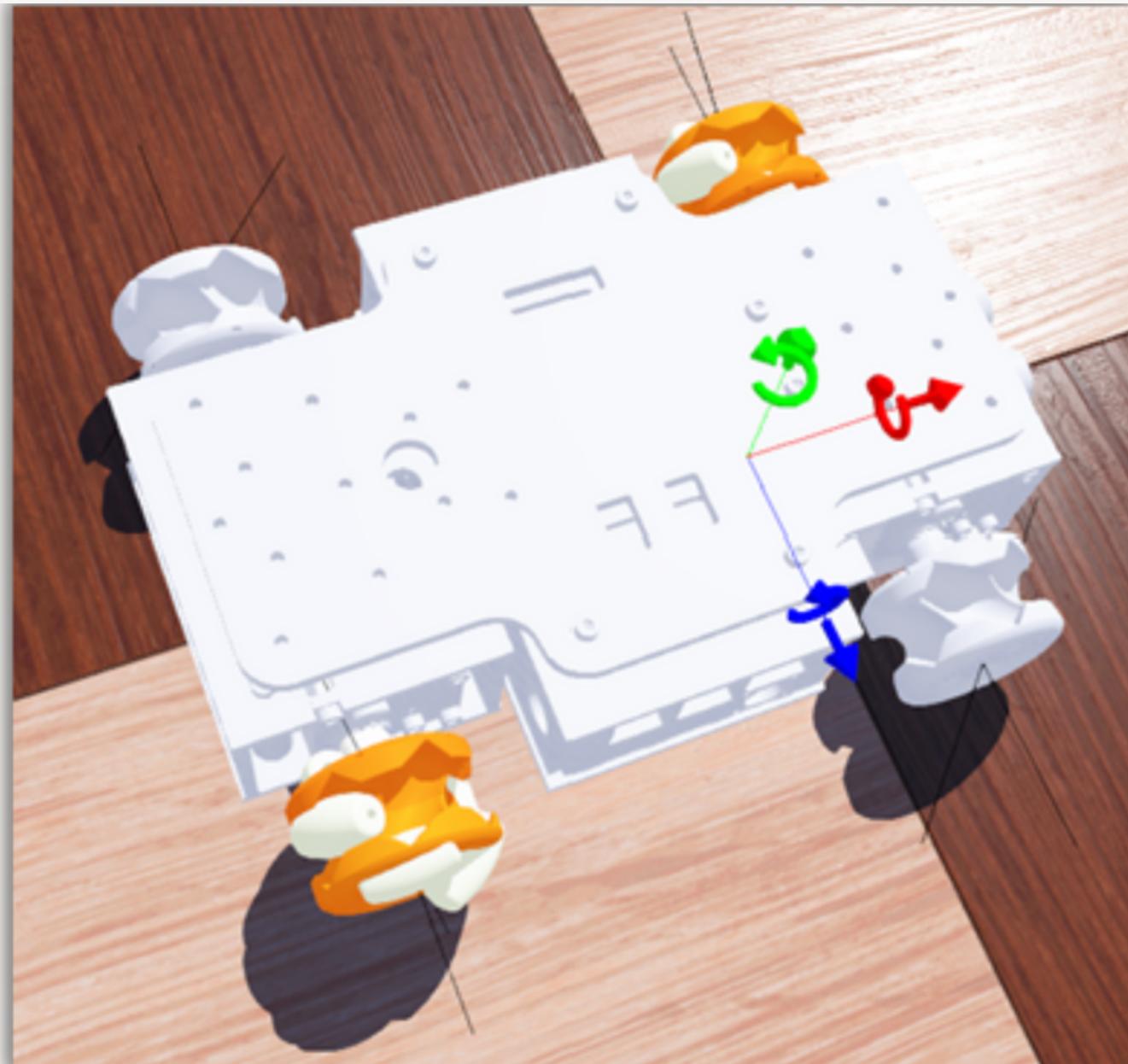


Video de los resultados





Conclusiones



Alternativa youBot KUKA



15.

ROBOT OMNIDIRECCIONAL "ASTÉRIX" | ABRIL DE 2021

Adecuaciones de diseño



- synchronization TRUE
- ▼ ● bodySlot
 - > ● InertialUnit "angulo"
 - > ● DEF pen Pen
 - > ● Compass "compass"
 - > ● GPS "gps"

```

int main(int argc, char **argv) {
    wb_robot_init();
    double zhd,xhd,t,a;
    //double angle;

    a=2.0;
    base_init();
    //arm_init();
    //gripper_init();
    base_goto_init(TIME_STEP);
    passive_wait(1.0);

    pen = wb_robot_get_device("pen");
    wb_pen_write(pen,true);
    int rojo = rand() & 0xff;
    wb_pen_set_ink_color(pen,rojo,0.9);

    while(true){
        step();
        t = 0.01*wb_robot_get_time();

        //Lemiscata
        //zhd=a*cos(t)/(1+sin(t)*sin(t));
        //xhd=(a*sin(t)*cos(t))/(1+sin(t)*sin(t));
        //angle = atan2(xhd,zhd);
        //Corazón
        zhd=(13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t))/10;
        xhd=(12*sin(t)-4*sin(3*t))/10;

        base_goto_set_target(-xhd,zhd,0);
        base_goto_run();
    }
    wb_robot_cleanup();
    return 0;
}

```

Aplicación del control de trayectoria

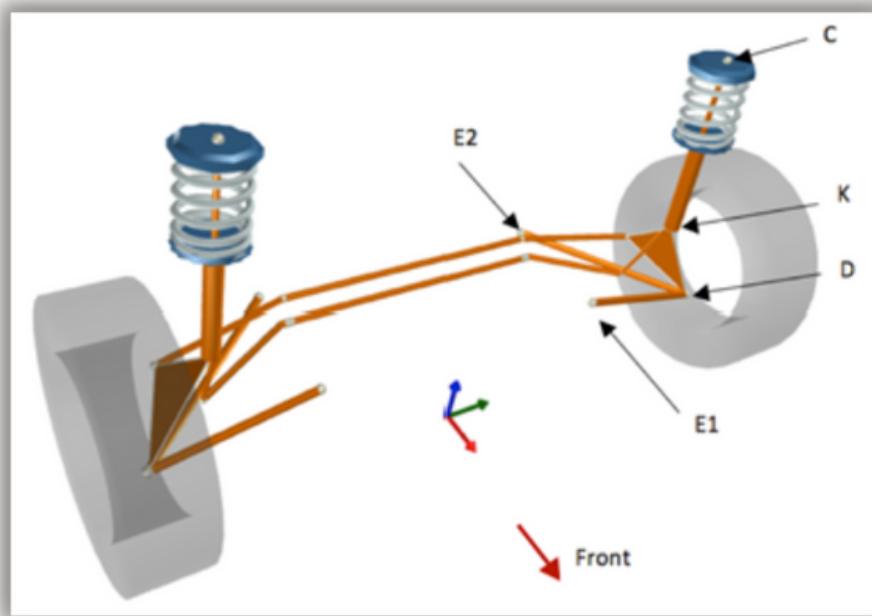
Mediante el uso de la librerías disponibles para el robot de KUKA es relativamente sencillo aplicar el control de trayectoria. Basta con asignar el punto deseado a la función *base_goto_set_target* y ejecutar la instrucción *base_goto_run()* y el robot se dirigirá al punto deseado

```

static void base_set_wheel_speeds_helper(double speeds[4]) {
    int i;
    for (i = 0; i < 4; i++)
        base_set_wheel_velocity(wheels[i], speeds[i]);
}

```

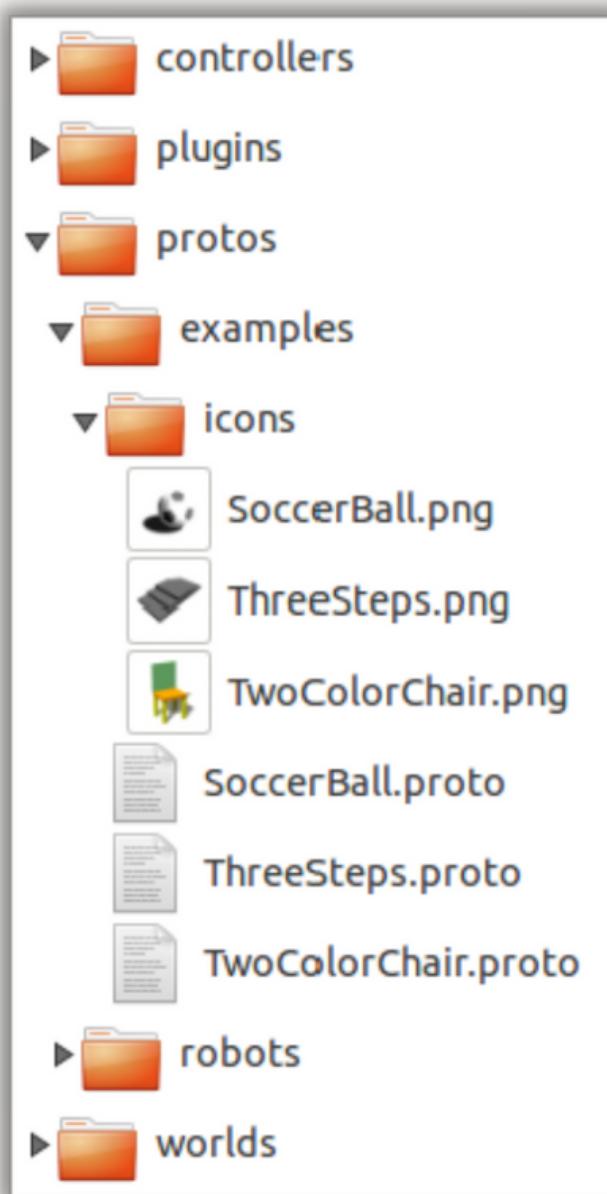
Uso de la suspensión para evitar vibraciones



- `suspensionSpringConstant 3`
- `suspensionDampingConstant 0`
- `suspensionAxis0 1 0`

Datos interesantes

Archivos PROTO



The screenshot shows a software interface with two main panes displaying hierarchical data structures:

- WorldInfo** node (highlighted in blue):
 - info
 - title ""
 - window ""
 - gravity 9.81
 - CFM 1e-05
 - ERP 0.2
 - physics ""
 - basicTimeStep 32
 - FPS 60
 - optimalThreadCount 1
 - physicsDisableTime 1
 - physicsDisableLinearThreshold 0.01
 - physicsDisableAngularThreshold 0.01
 - defaultDamping NULL
 - inkEvaporation 0
 - coordinateSystem "NUE"
 - gpsCoordinateSystem "local"
 - gpsReference 0 0 0
 - lineScale 0.1
 - randomSeed 0
 - contactProperties
- ContactProperties** node (highlighted in blue):
 - material1 "default"
 - material2 "default"
 - coulombFriction
 - frictionRotation 0 0
 - bounce 0.5
 - bounceVelocity 0.01
 - forceDependentSlip
 - softERP 0.2
 - softCFM 0.001
 - bumpSound "sounds/bump.wav"
 - rollSound "sounds/roll.wav"
 - slideSound "sounds/slide.wav"

Configuración del contacto entre dos materiales



Referencias

1

[1] Webots.
<http://www.cyberbotics.com>.
Open-source Mobile Robot
Simulation Software.

2

Robot omnidireccional.
[Wikitronica.labc.usb.ve](http://wikitronica.labc.usb.ve).
(2015). Recuperado el 5 de
abril de 2021, Disponible en:
<http://wikitronica.labc.usb.ve/index.php/Robot>.

3

Robot con Mecanum Wheel
controlado por Arduino. Ingeniería,
Informática y Diseño. (2019).
Recuperado el 6 de abril de 2021,
Disponible en:
<https://www.luisllamas.es/robot-con-mecanum-wheel-controlado-por-arduino>

4

IL, B. (2021). Una Introducción a
los Robots Móviles. Aadeca.org.
Recuperado el 7 de abril de 2021,
Disponible en:
https://www.aadeca.org/pdf/CP_monografias/monografia_robot_movil.pdf.