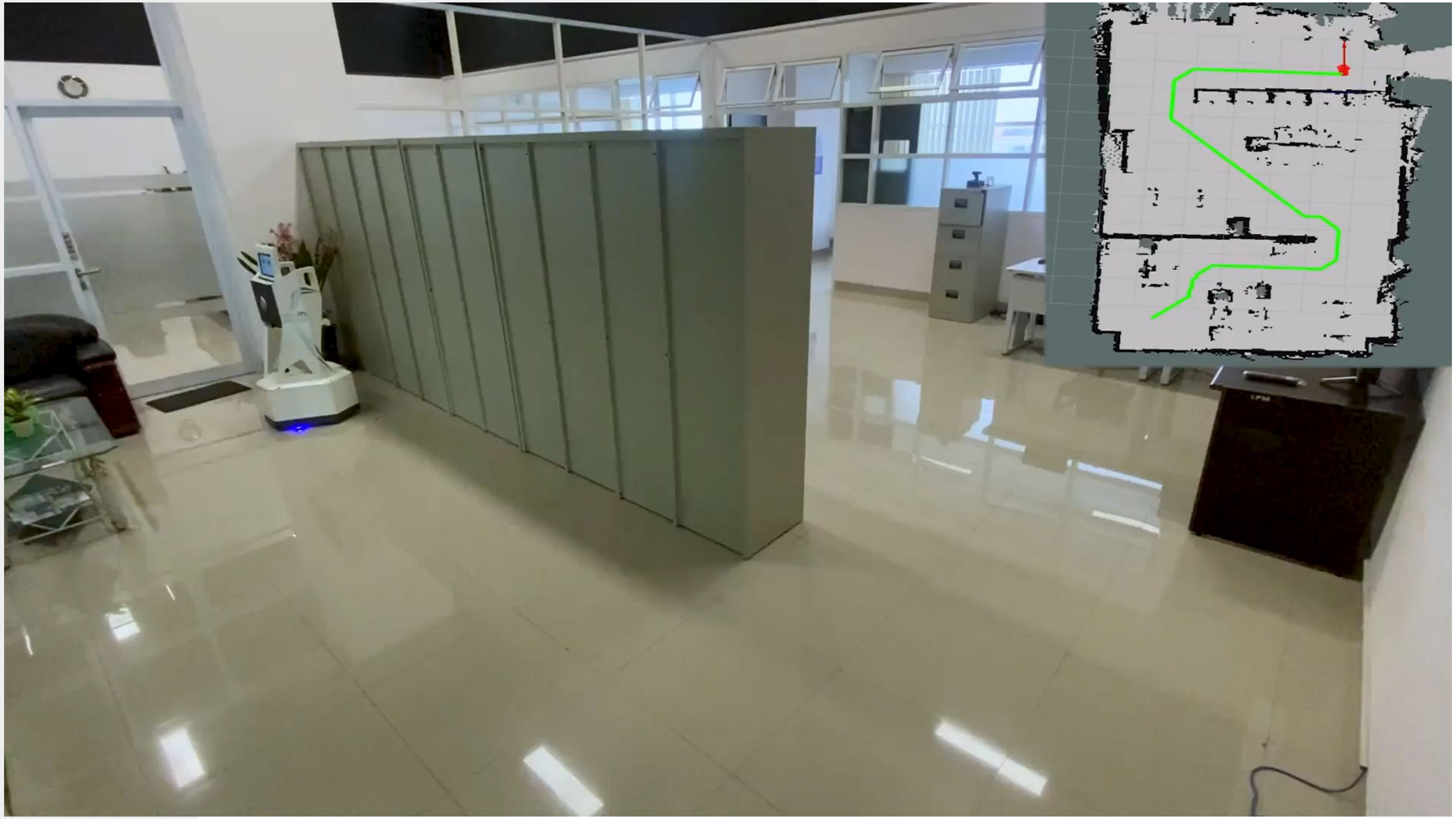
INTRODUCTION TO AUTONOMOUS ROBOT AND ROBOT NAVIGATION

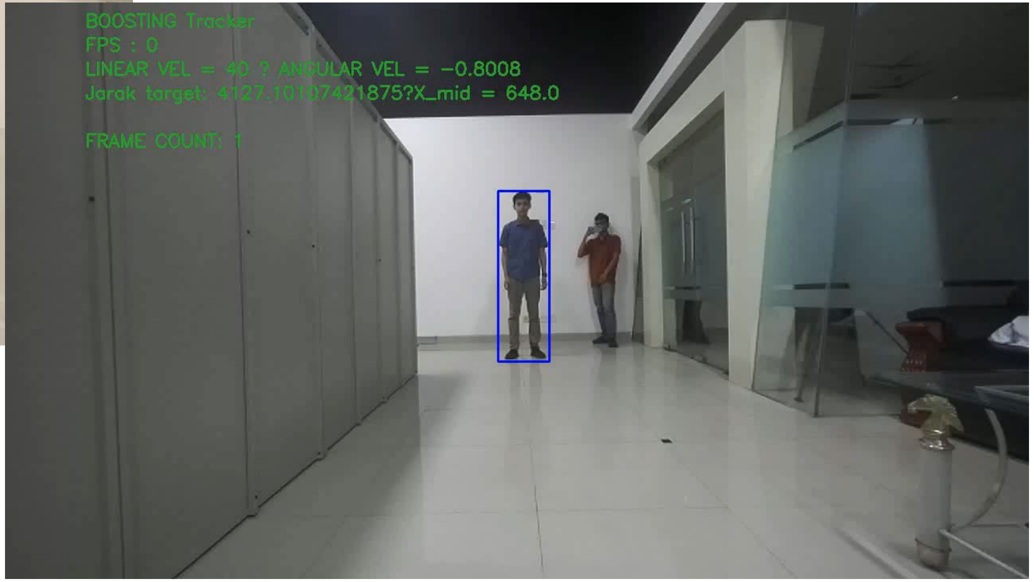
Djoko Purwanto
Dept. of Electrical Engineering, ITS
djoko@its.ac.id

INTRODUCTION

- Path Tracking



- Person Follower



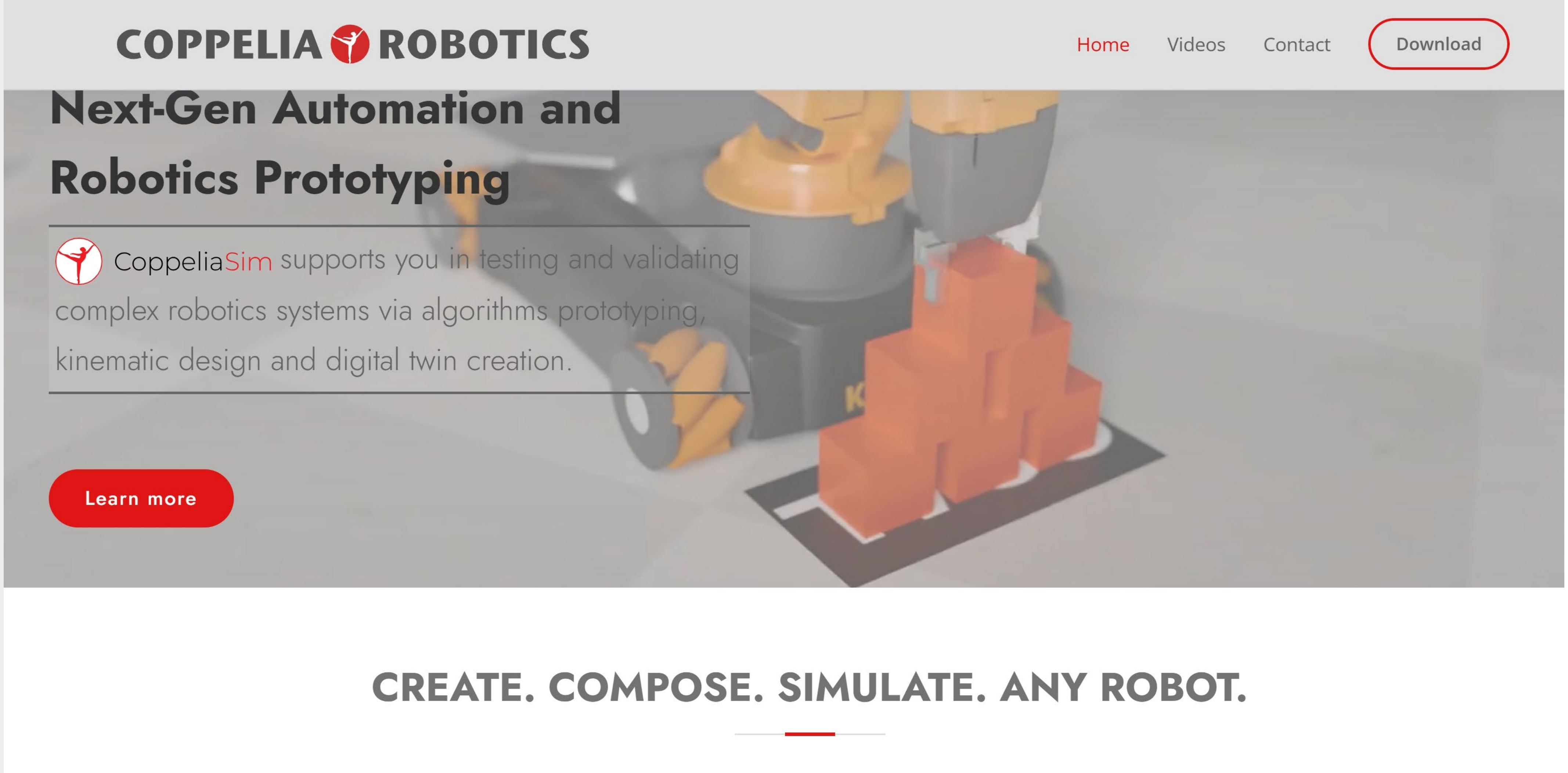
```
BOOSTING Tracker
FPS : 0
LINEAR VEL = 40 ? ANGULAR VEL = -0.8008
Jarak target: 4127.10107421875?X_mid = 648.0

FRAME COUNT: 1
```

(1) Robot Applications, (2) Robot Navigation

(1)	6 warehouse robots that are reshaping the industry	https://www.youtube.com/watch?v=LDhJ5I89H_I
	iFollow : Collaborative Robots for logistics	https://www.youtube.com/watch?v=Jkv9qeYFtPs
	Security Robot (Robot guard) Rover S5	https://www.youtube.com/watch?v=ZprJHzpmsLk
	Rise of autonomous robots amid COVID-19 outbreak	https://www.youtube.com/watch?v=r-tfQH-r_3M
	iCar ITS, Persembahan ITS di Hari Peringatan Kemerdekaan Indonesia ke-75	https://www.youtube.com/watch?v=I2XRN4_ISyY
(2)	Clever Autonomy for Mobile Robots - KUKA Navigation Solution	https://www.youtube.com/watch?v=kN9a7W_hnSQ

ROBOT SIMULATOR

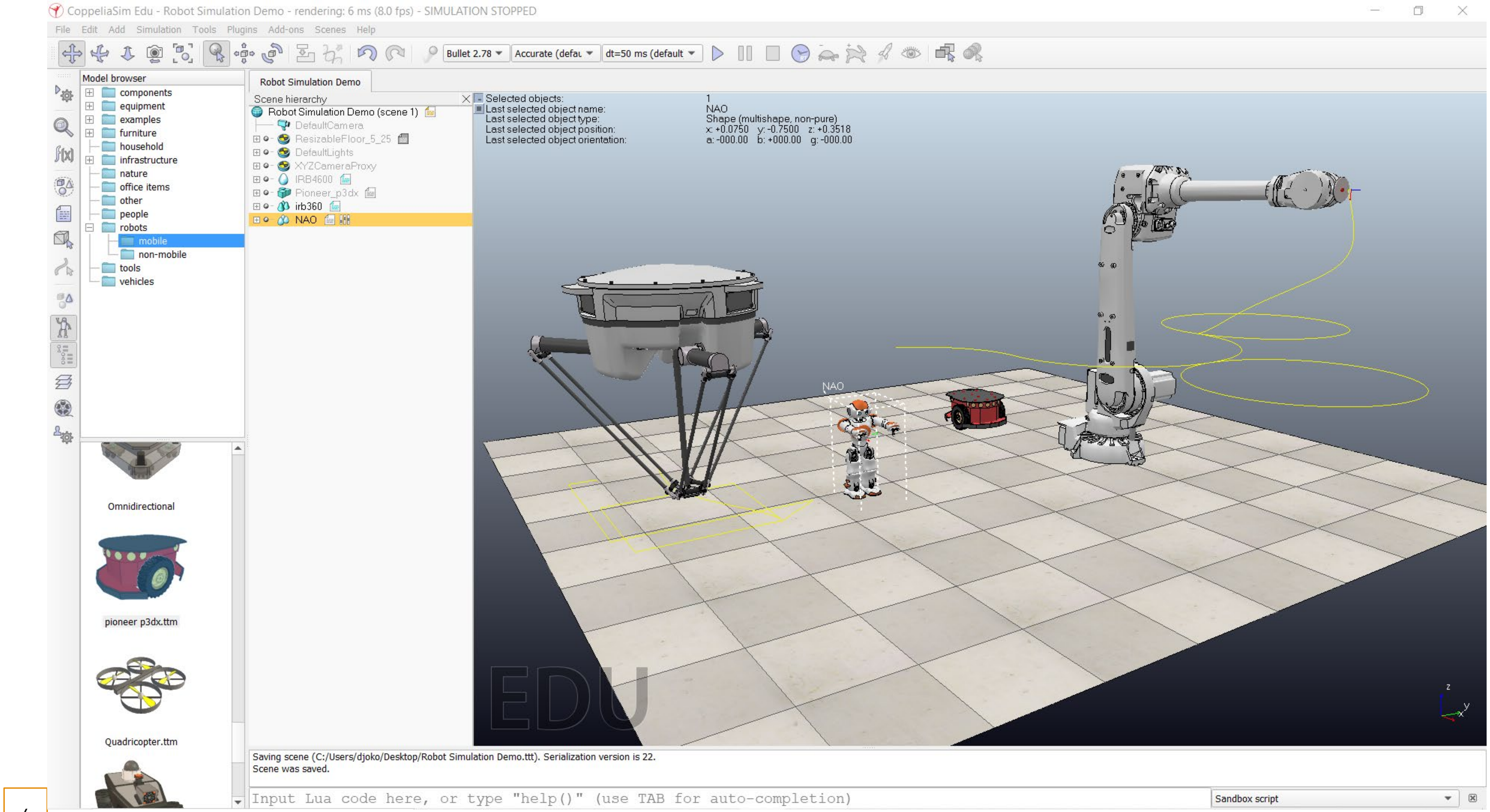
https://www.coppeliarobotics.com/



Next-Gen Automation and Robotics Prototyping

CoppeliaSim supports you in testing and validating complex robotics systems via algorithms prototyping, kinematic design and digital twin creation.

CREATE. COMPOSE. SIMULATE. ANY ROBOT.



CoppeliaSim Edu - Robot Simulation Demo - rendering: 6 ms (8.0 fps) - SIMULATION STOPPED

Bullet 2.78 | Accurate (defau | dt=50 ms (default

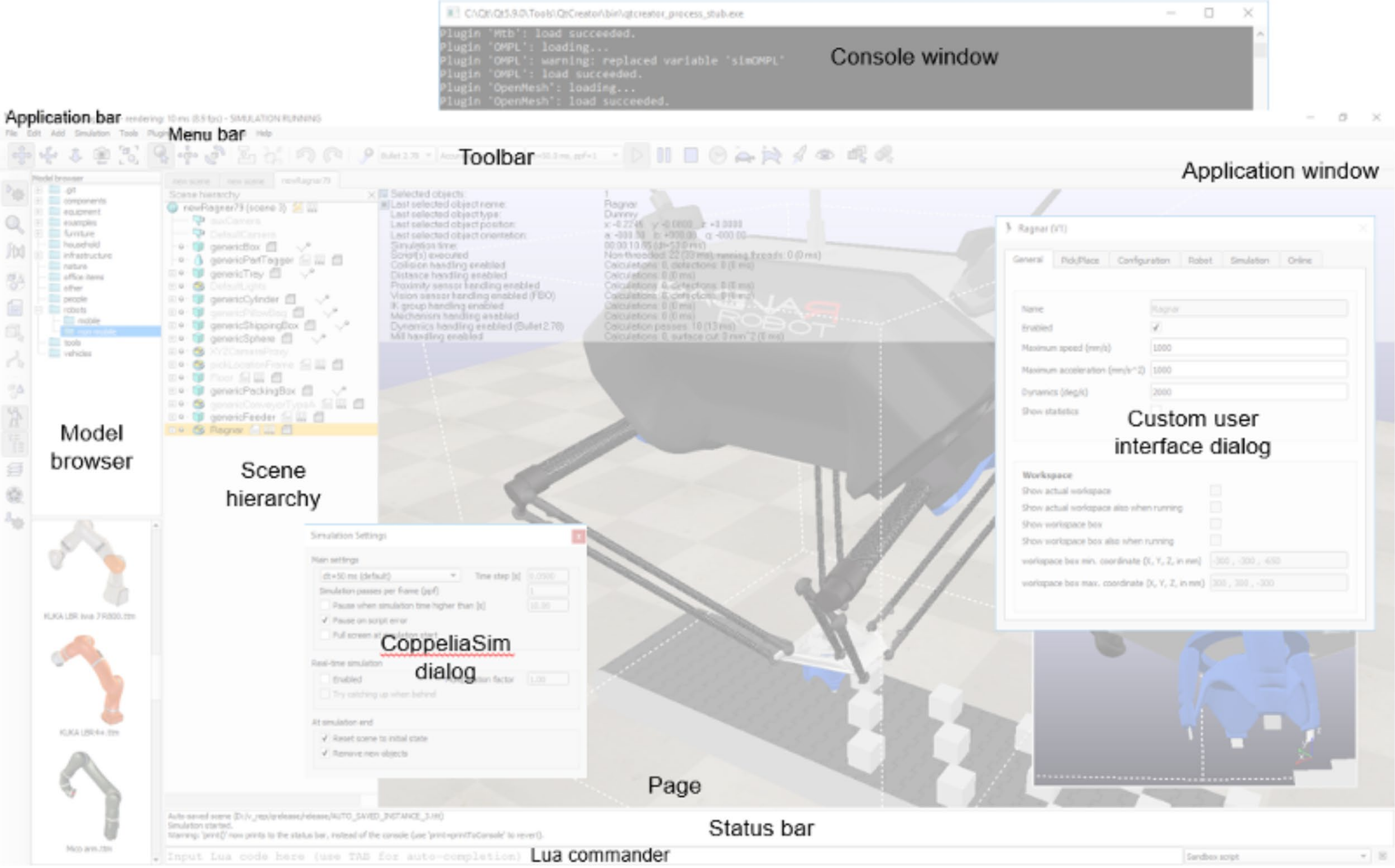
Selected objects: 1
Last selected object name: NAO
Last selected object type: Shape (multishape, non-pure)
Last selected object position: x: +0.0750 y: -0.7500 z: +0.3518
Last selected object orientation: a: -000.00 b: +000.00 g: -000.00

```
Saving scene (C:/Users/djoko/Desktop/Robot Simulation Demo.ttt). Serialization version is 22.
Scene was saved.
```

```
Input Lua code here, or type "help()" (use TAB for auto-completion)
```

Sandbox script

UI ELEMENTS

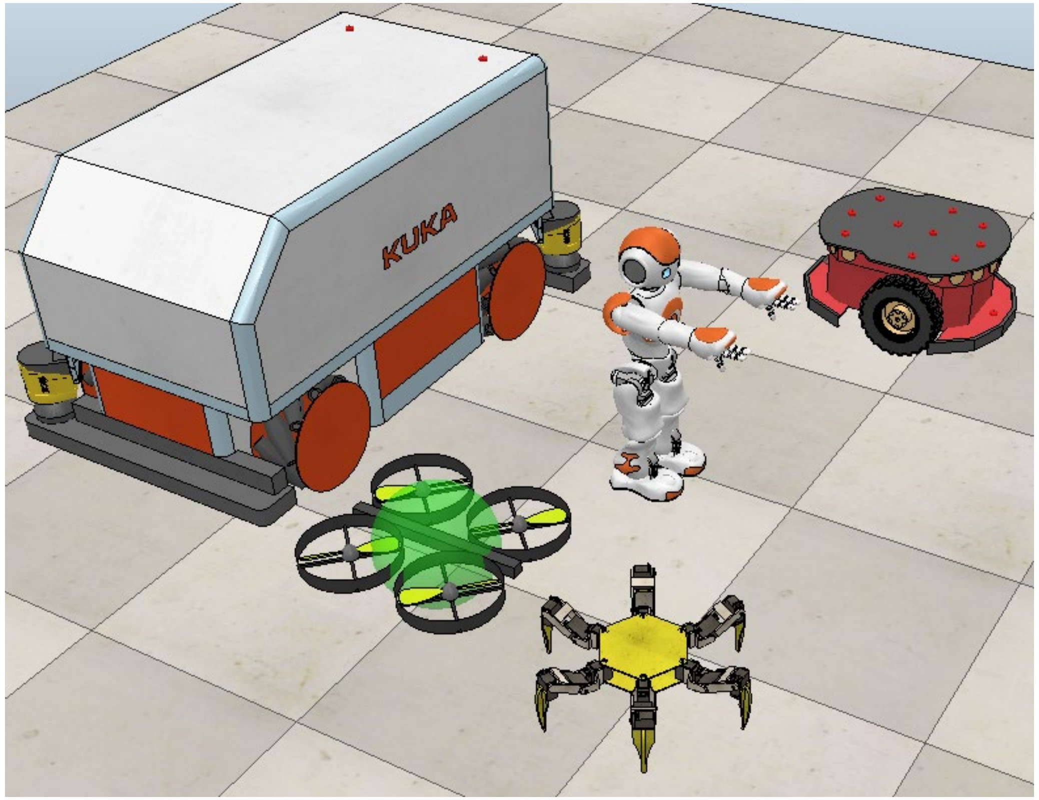


```
Plugin 'Mtb': load succeeded.
Plugin 'OMPL': loading...
Plugin 'OMPL': warning: replaced variable 'simOMPL'
Plugin 'OMPL': load succeeded.
Plugin 'OpenMesh': loading...
Plugin 'OpenMesh': load succeeded.
```

C:\Qt\Qt5.9.0\Tools\QtCreator\bin\qtcreator_process_stub.exe
Console window
Application bar
Menu bar
Toolbar
Application window
Model browser
Scene hierarchy
Custom user interface dialog
CoppeliaSim dialog
Page
Status bar
Lua commander
Input Lua code here (use TAB for auto-completion)

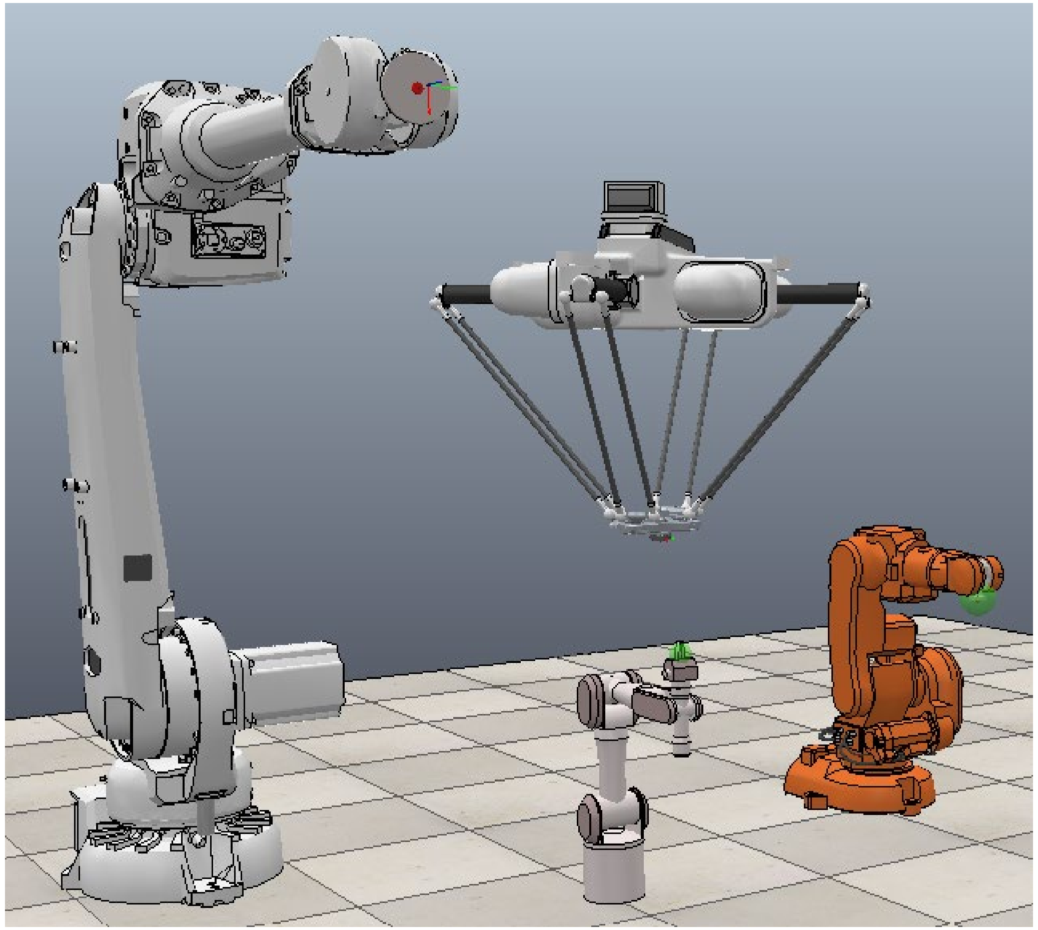
MOBILE ROBOTS

- Omni Wheels
- Differential Wheels
- Humanoid
- Hexapod
- Quadcopter



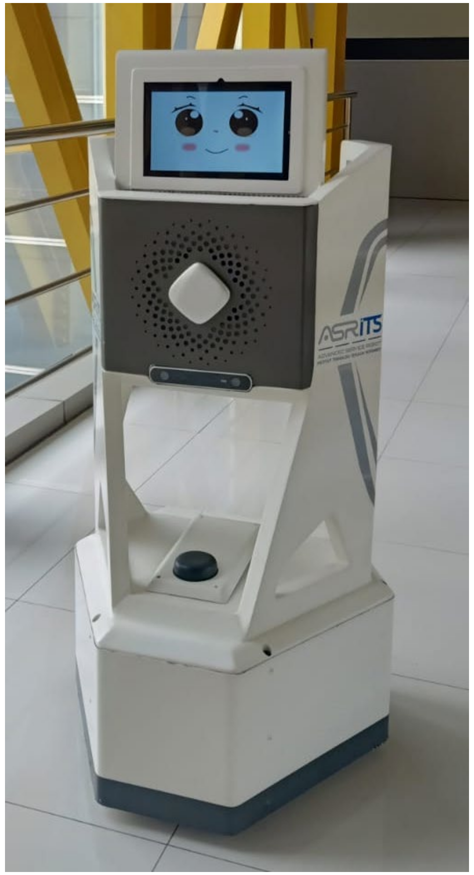
FIXED ROBOTS

- Industrial Robot
- Parallel Robot



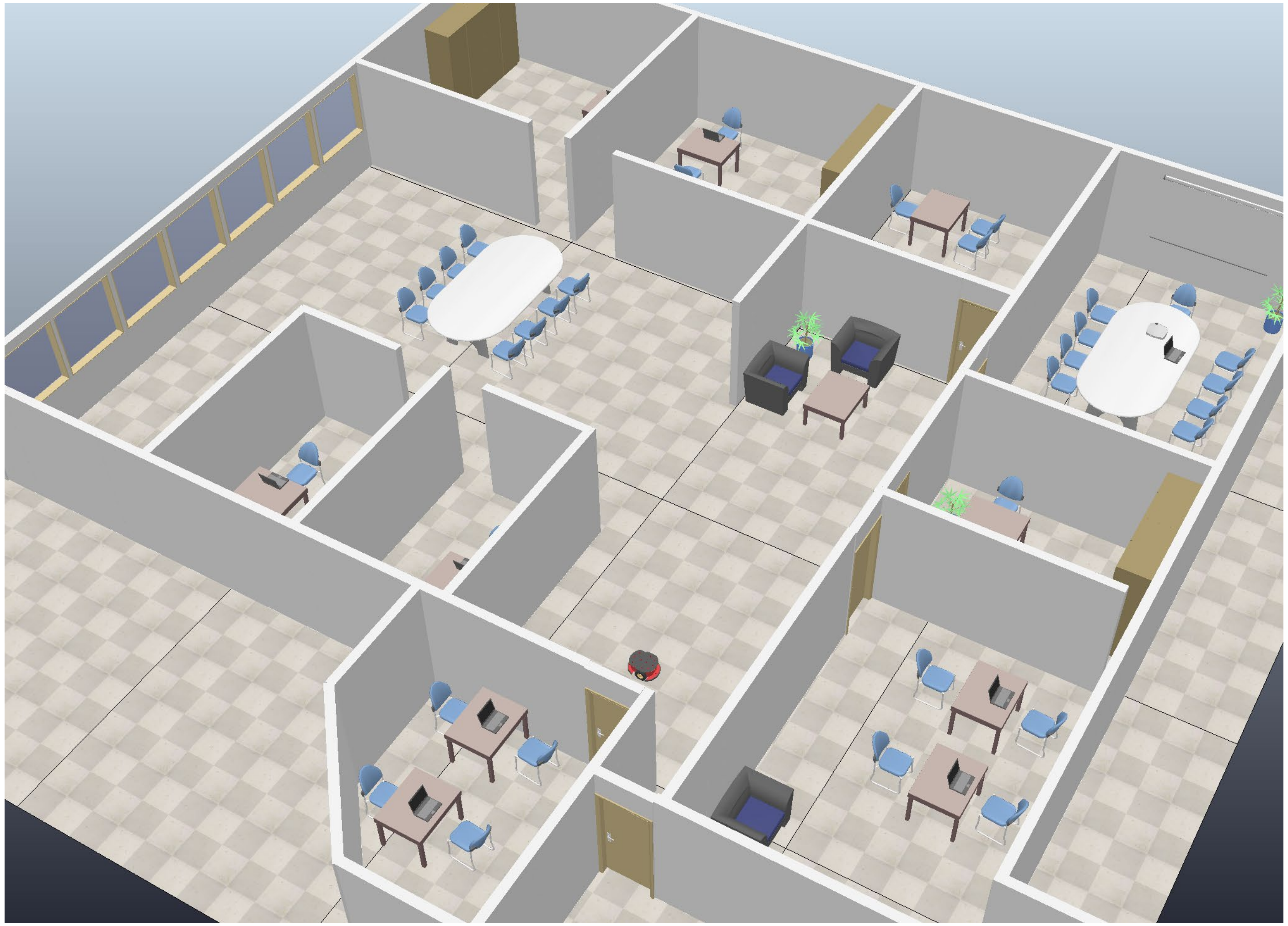
CUSTOM ROBOTS

- ASRITS (Advanced Service Robot of ITS)

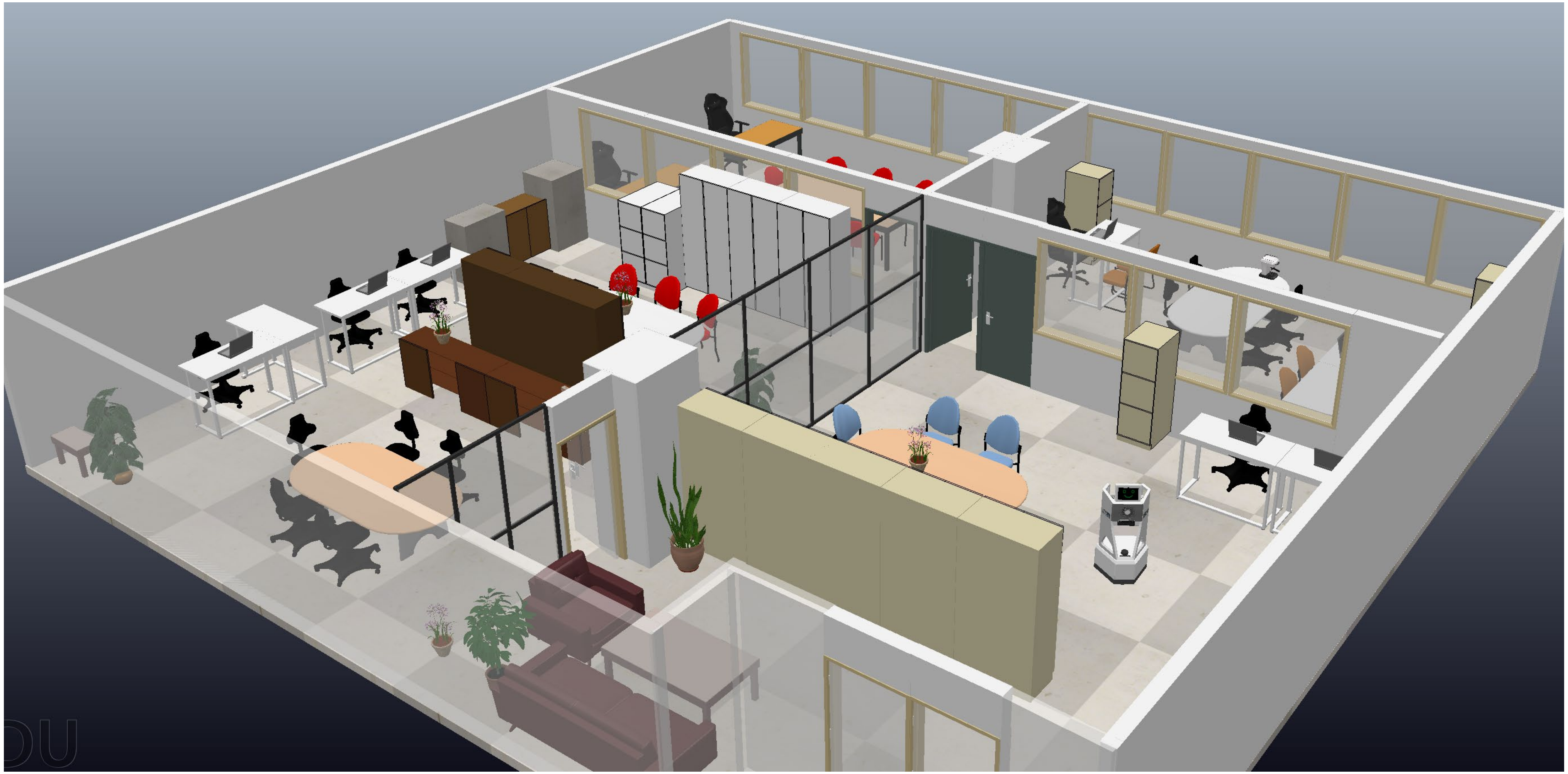


SIMULATION ENVIRONMENT

- Office

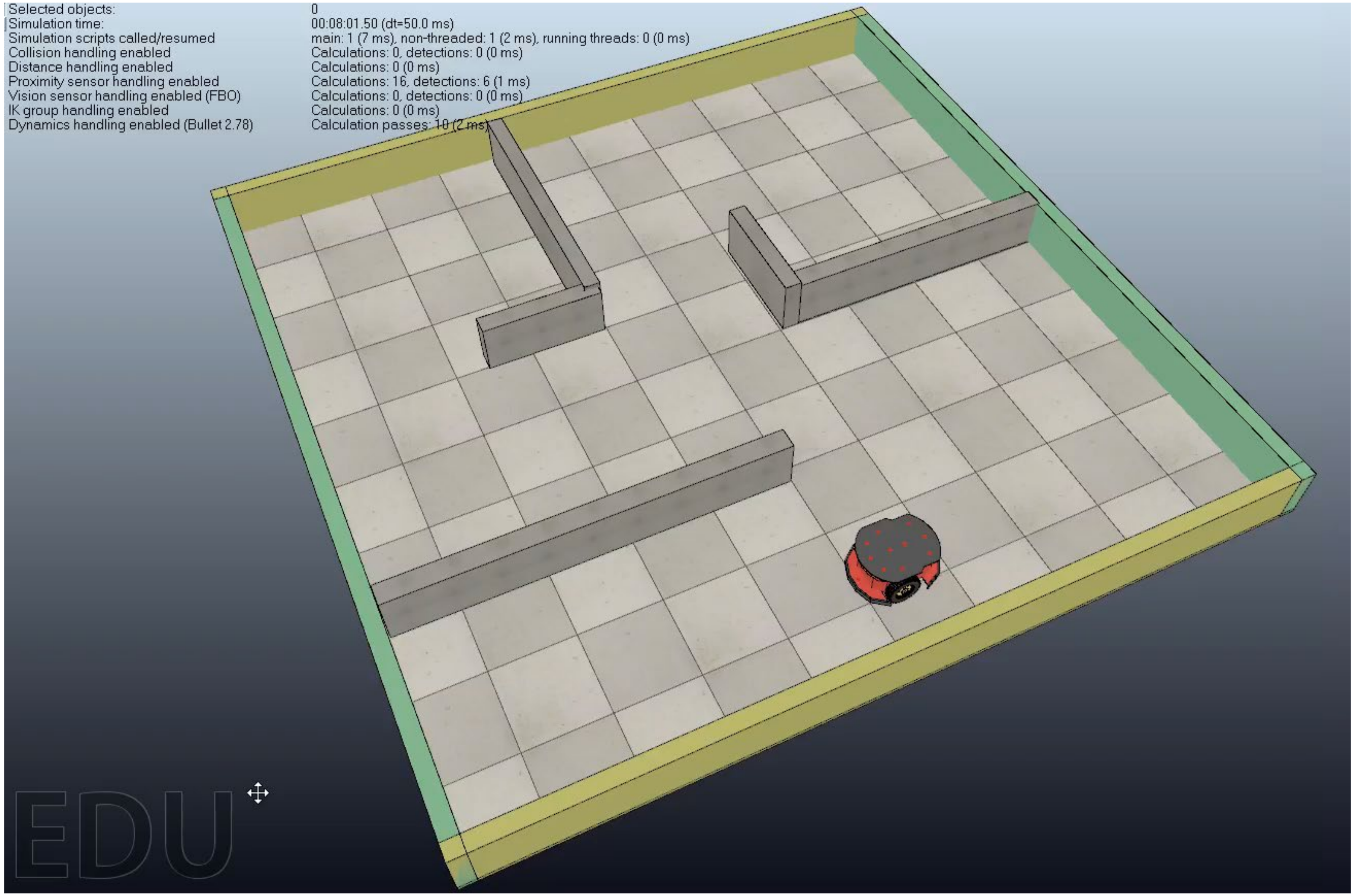


- Research Center of Artificial Intelligence and Health Technology - ITS



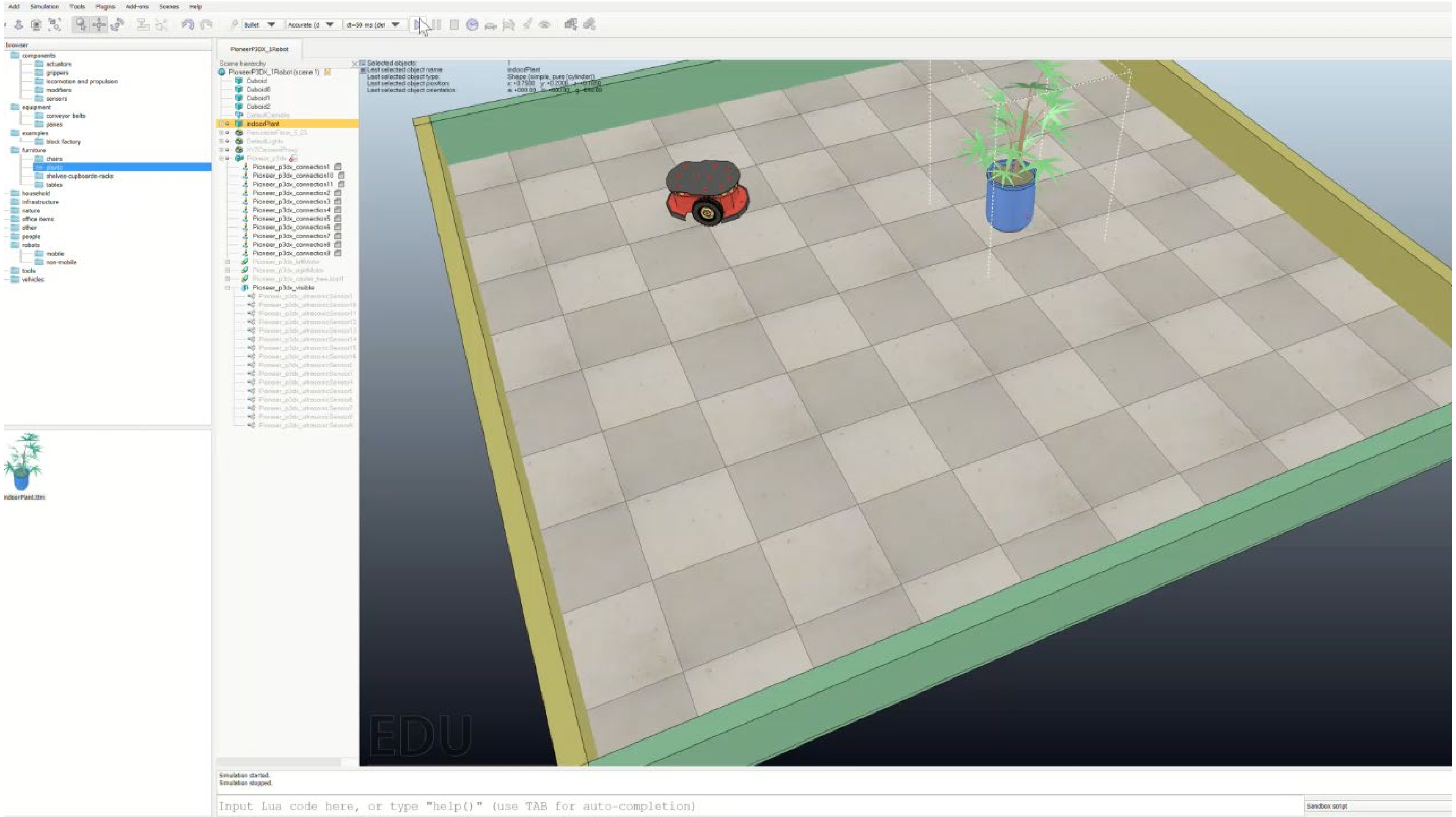
Simulation Examples

- Room Inspection



```
Selected objects:                          0
Simulation time:                           00:08:01.50 (dt=50.0 ms)
Simulation scripts called/resumed          main: 1 (7 ms), non-threaded: 1 (2 ms), running threads: 0 (0 ms)
Collision handling enabled                 Calculations: 0, detections: 0 (0 ms)
Distance handling enabled                  Calculations: 0 (0 ms)
Proximity sensor handling enabled          Calculations: 16, detections: 6 (1 ms)
Vision sensor handling enabled (FBO)       Calculations: 0, detections: 0 (0 ms)
IK group handling enabled                  Calculations: 0 (0 ms)
Dynamics handling enabled (Bullet 2.78)    Calculation passes: 10 (2 ms)
```

- Object Follower



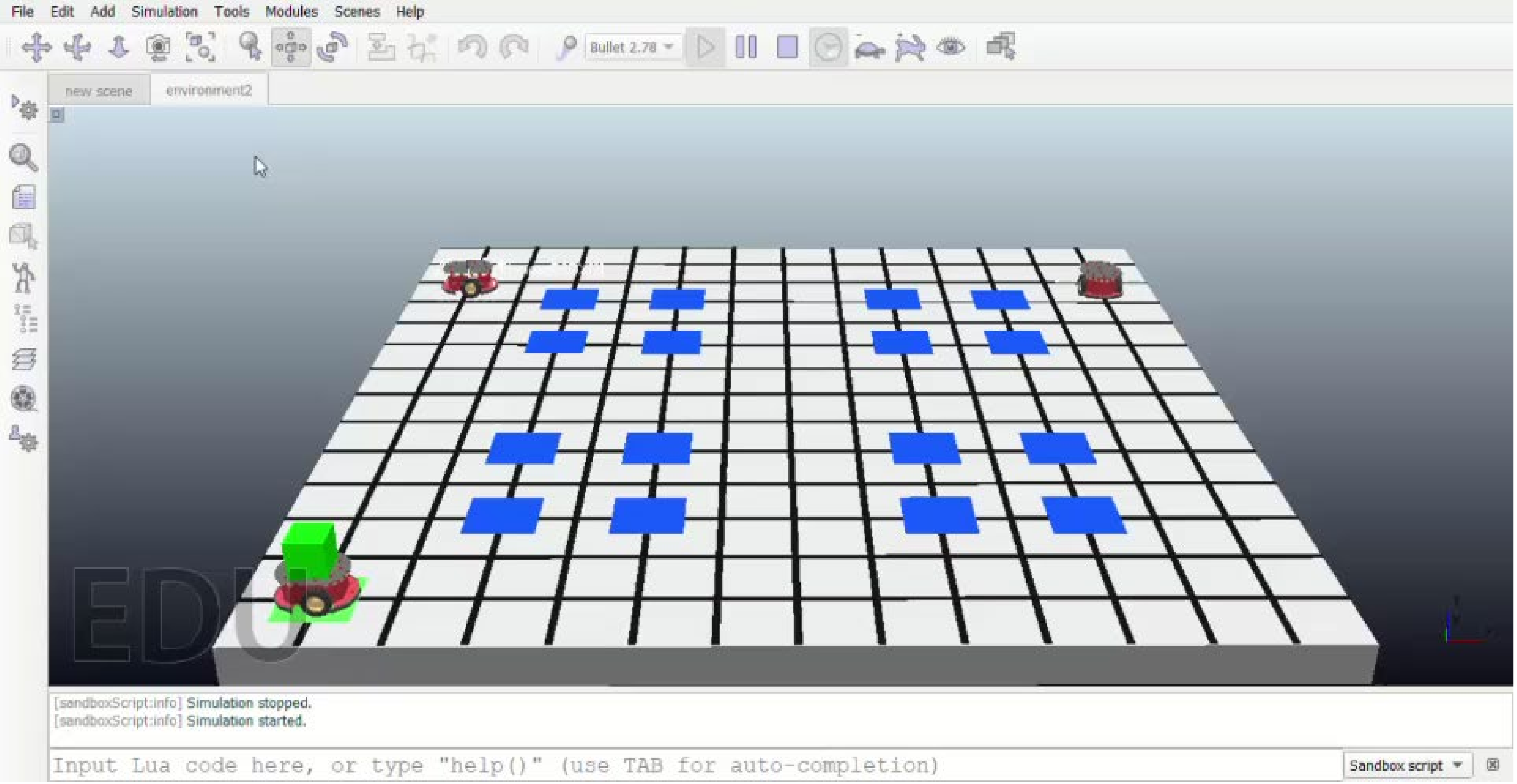
Bullet | Accurate (d | dt=50 ms (def

```
Selected objects:                    1
Last selected object name            indoorPlant
Last selected object type            Shape (simple, pure (cylinder))
Last selected object position        x: +0.7500   y: +0.2000   z: +0.1050
Last selected object orientation     a: +000.00   b: +000.00   g: +000.00
```

```
Simulation started.
Simulation stopped.
```

Input Lua code here, or type "help()" (use TAB for auto-completion)

- Warehouse

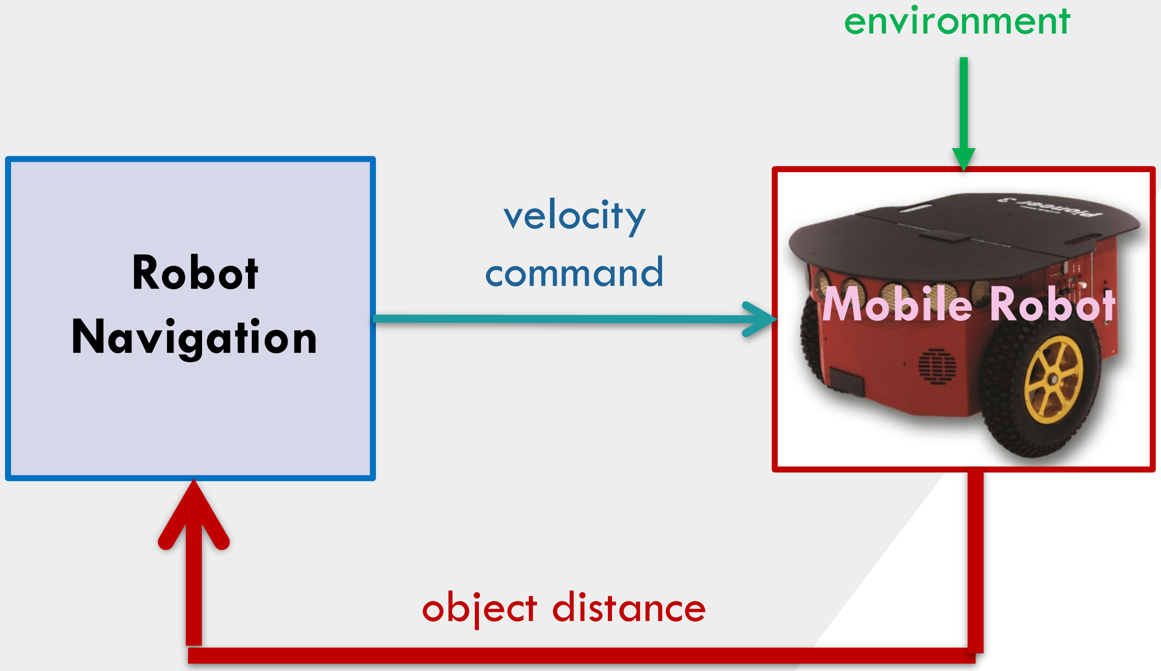


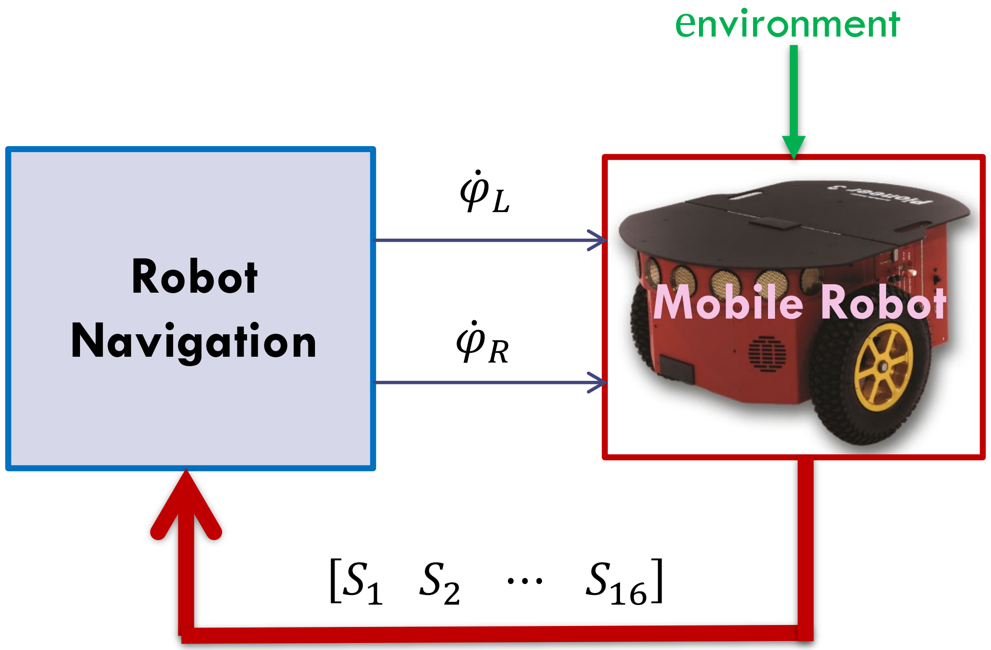
```
[sandboxScript:info] Simulation stopped.
[sandboxScript:info] Simulation started.
```

```
Input Lua code here, or type "help()" (use TAB for auto-completion)
```

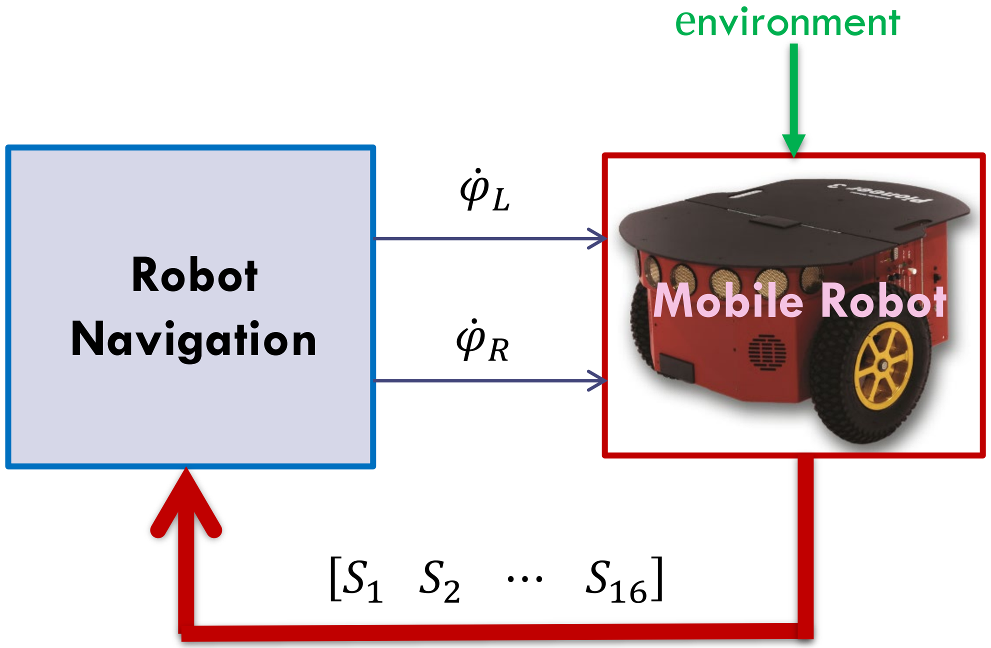
Sandbox script

ROBOT NAVIGATION





Braitenberg Algorithm Based Robot Navigation



Sensor Data Processing

$$D_n = \begin{cases} 1 - \dfrac{d_n - DD\text{max}}{DD\text{no} - DD\text{max}} & S_n < DD\text{no} \\ 0 & \text{otherwise} \end{cases} \qquad n = 1,2,\cdots,16$$

$$d_n = \begin{cases} DD\text{max} & S_n < DD\text{max} \\ S_n & \text{otherwise} \end{cases} \qquad n = 1,2,\cdots,16$$

$DD\text{no}$: No detection distance

$DD\text{max}$: Maximum detection distance

Braitenberg Algorithm

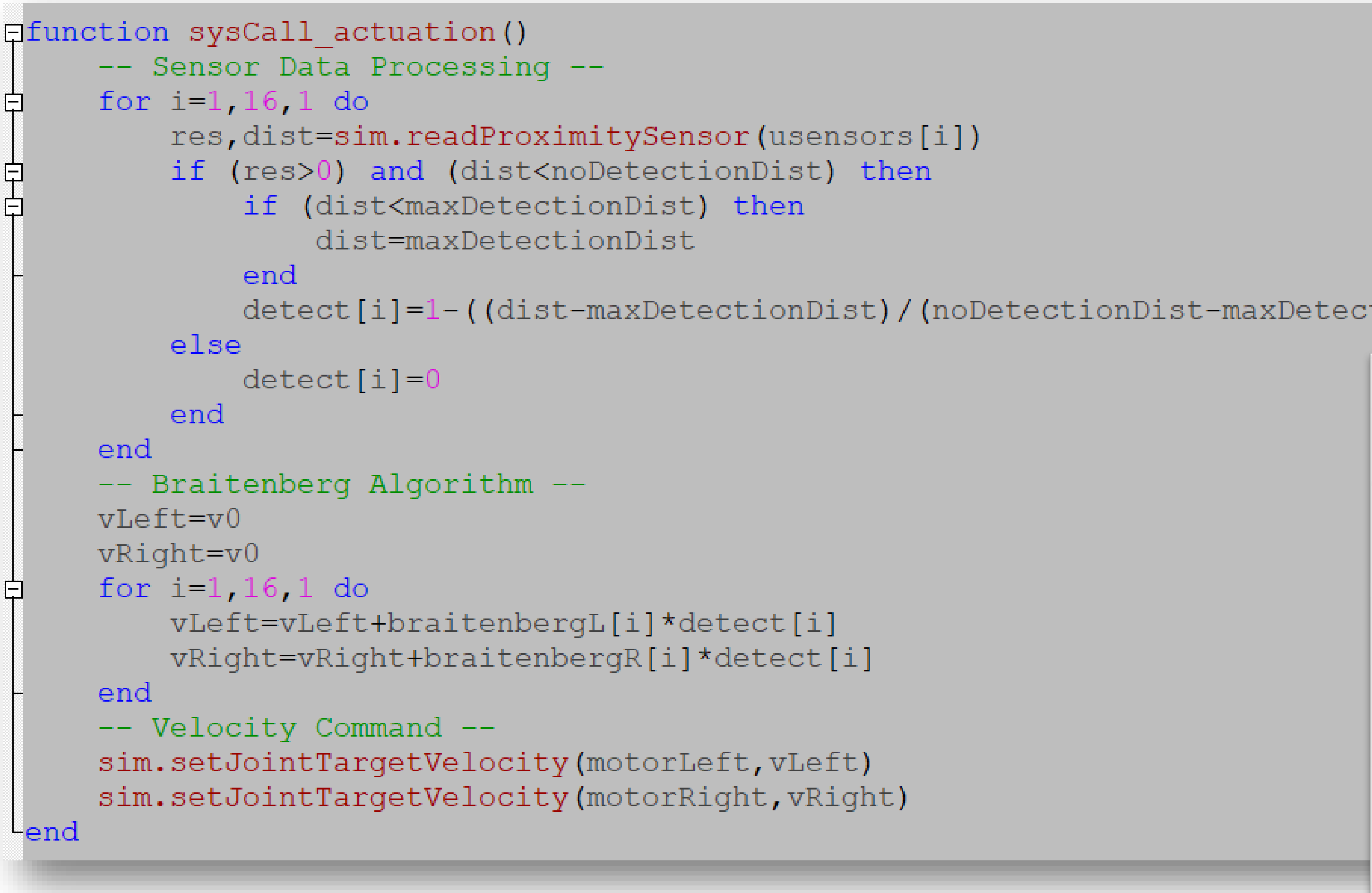
Left angular velocity : $\dot{\varphi}_L = \dot{\varphi}_0 + \sum_{n=1}^{16} BL_n \times D_n$

Right angular velocity : $\dot{\varphi}_R = \dot{\varphi}_0 + \sum_{n=1}^{16} BR_n \times D_n$

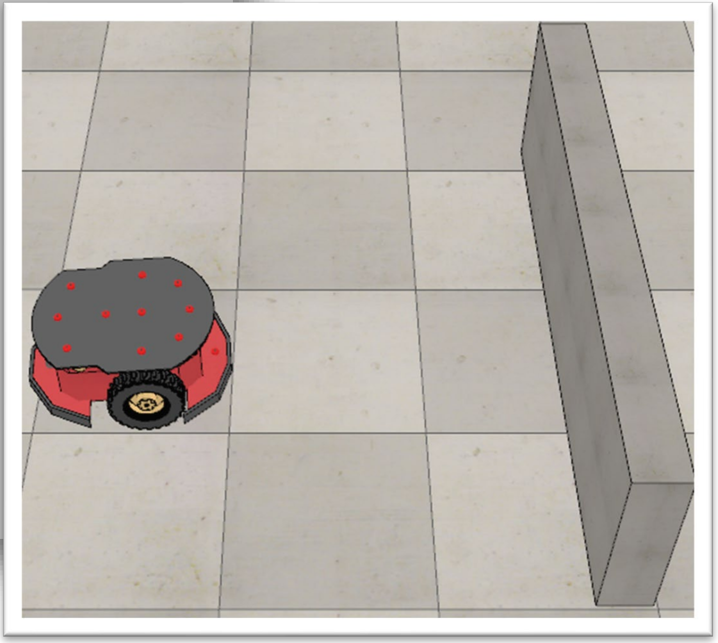
$\dot{\varphi}_0$: initial angular velocity

BL, BR : Left, Right Braitenberg constant

Lua Programming



```lua
function sysCall_actuation()
    -- Sensor Data Processing --
    for i=1,16,1 do
        res,dist=sim.readProximitySensor(usensors[i])
        if (res>0) and (dist<noDetectionDist) then
            if (dist<maxDetectionDist) then
                dist=maxDetectionDist
            end
            detect[i]=1-((dist-maxDetectionDist)/(noDetectionDist-maxDetectionDist))
        else
            detect[i]=0
        end
    end
    -- Braitenberg Algorithm --
    vLeft=v0
    vRight=v0
    for i=1,16,1 do
        vLeft=vLeft+braitenbergL[i]*detect[i]
        vRight=vRight+braitenbergR[i]*detect[i]
    end
    -- Velocity Command --
    sim.setJointTargetVelocity(motorLeft,vLeft)
    sim.setJointTargetVelocity(motorRight,vRight)
end
```



Analysis of Robot Navigation

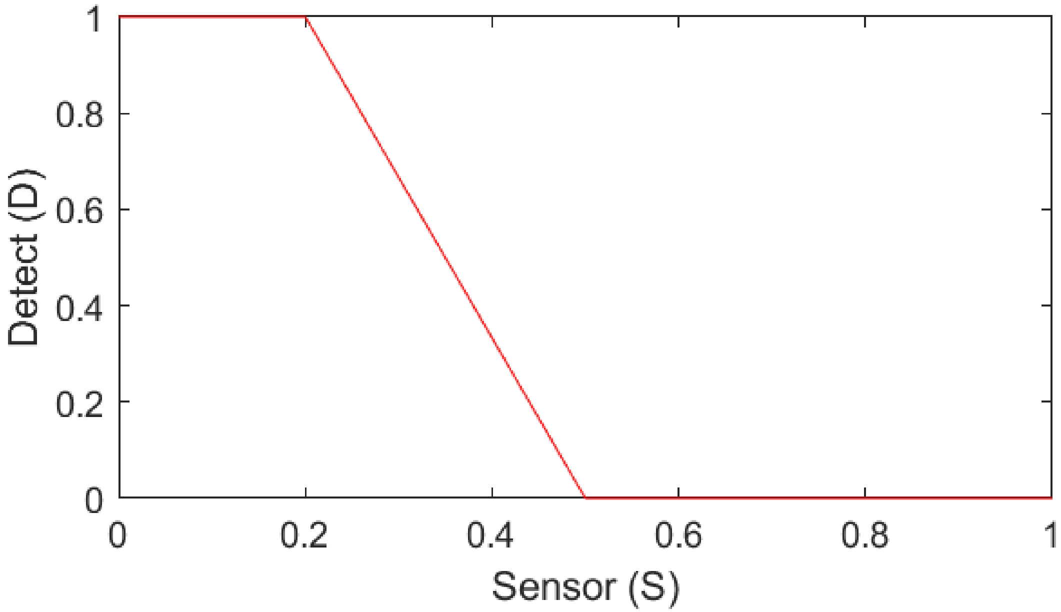
Sensor Data Processing

$$D_n = \begin{cases} 1 - \dfrac{d_n - DD\text{max}}{DD\text{no} - DD\text{max}} & S_n < DD\text{no} \\ 0 & \text{otherwise} \end{cases} \qquad n = 1,2,\cdots,16$$

$$d_n = \begin{cases} DD\text{max} & S_n < DD\text{max} \\ S_n & \text{otherwise} \end{cases} \qquad n = 1,2,\cdots,16$$

$DD\text{no} = 0.5$ (No detection distance)

$DD\text{max} = 0.2$ (Maximum detection distance)



Braitenberg Algorithm

Left angular velocity : $$\dot{\varphi}_L = \dot{\varphi}_0 + \sum_{n=1}^{16} BL_n \times D_n$$

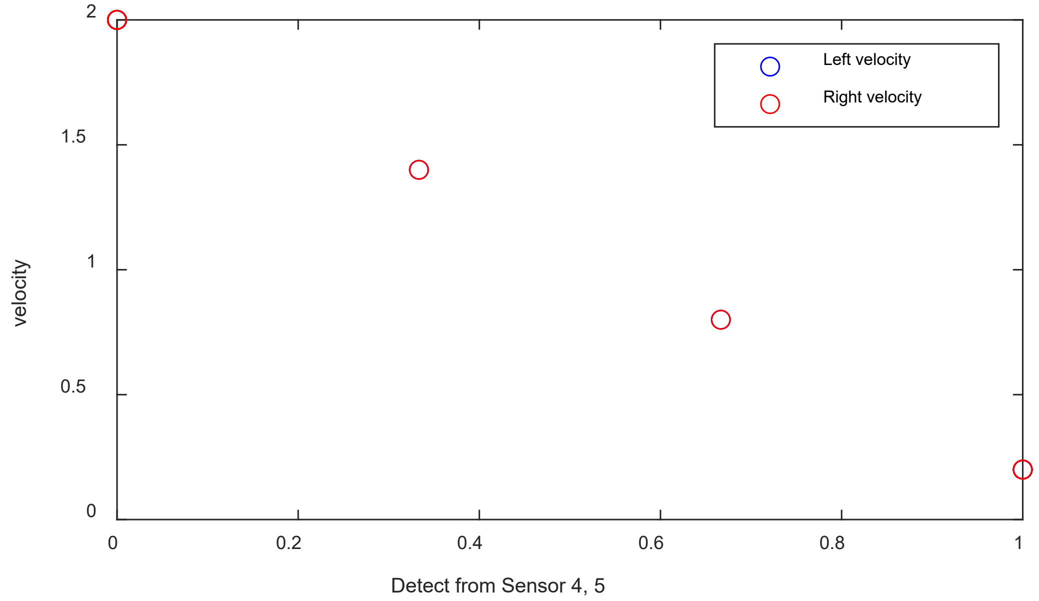
Right angular velocity : $$\dot{\varphi}_R = \dot{\varphi}_0 + \sum_{n=1}^{16} BR_n \times D_n$$

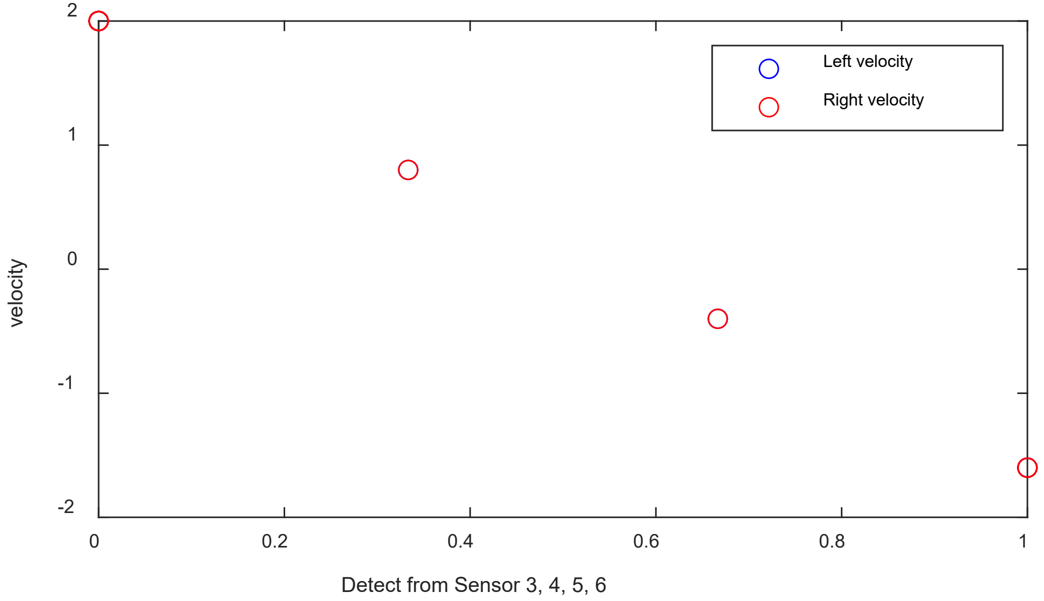
$\dot{\varphi}_0 = 2$ (initial angular velocity)

$BL = [-0.2\ \ -0.4\ \ -0.6\ \ -0.8\ \ -1\ \ -1.2\ \ -1.4\ \ -1.6\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$ (Left Braitenberg constant)

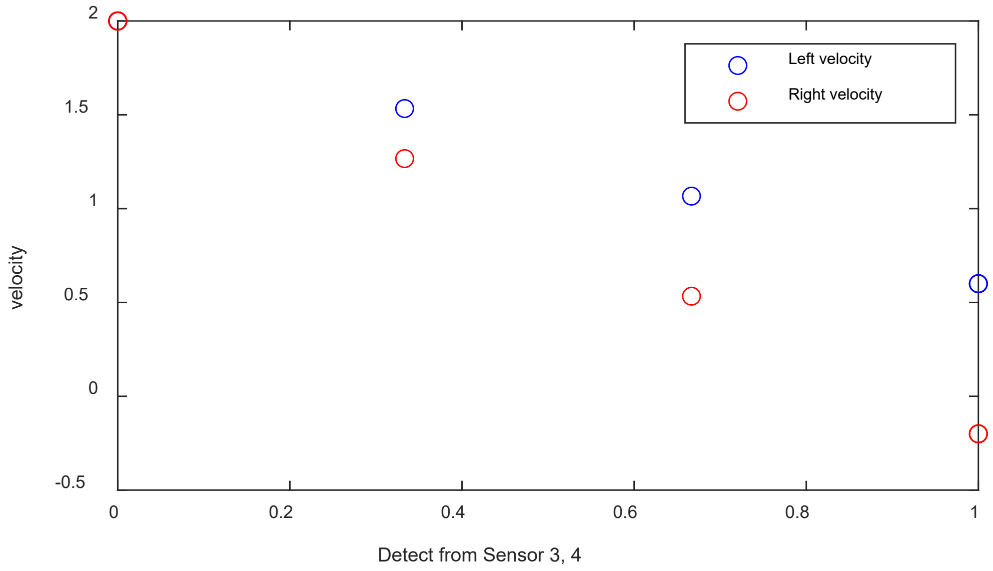
$BR = [-1.6\ \ -1.4\ \ -1.2\ \ -1\ \ -0.8\ \ -0.6\ \ -0.4\ \ -0.2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$ (Right Braitenberg constant)

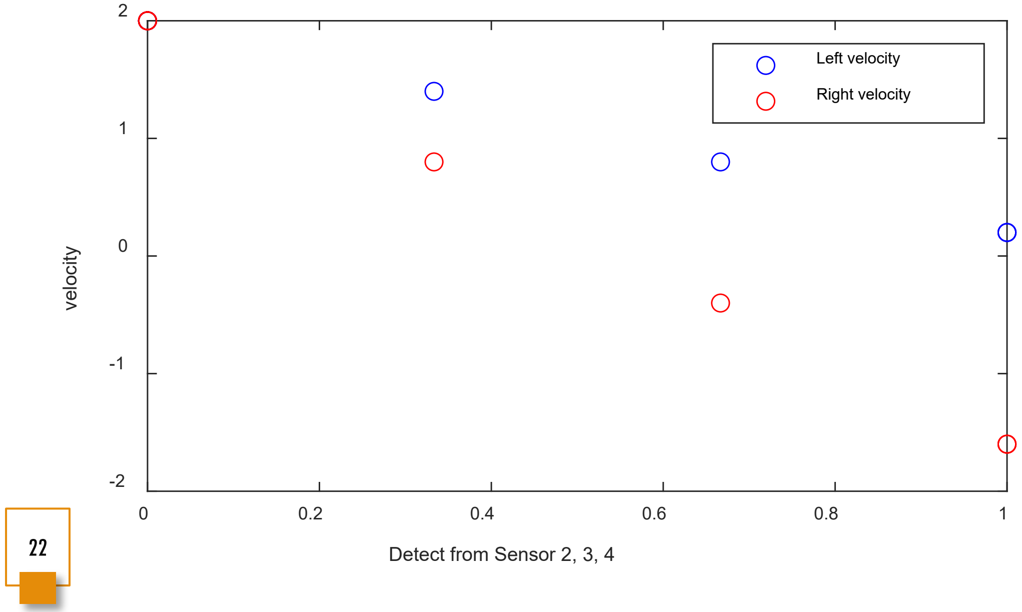
Obstacle at front direction



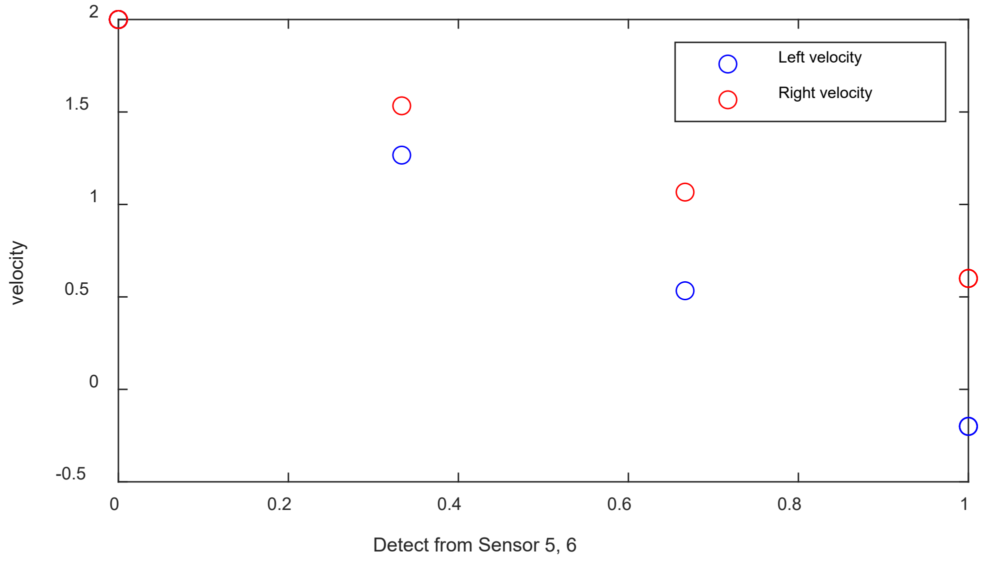


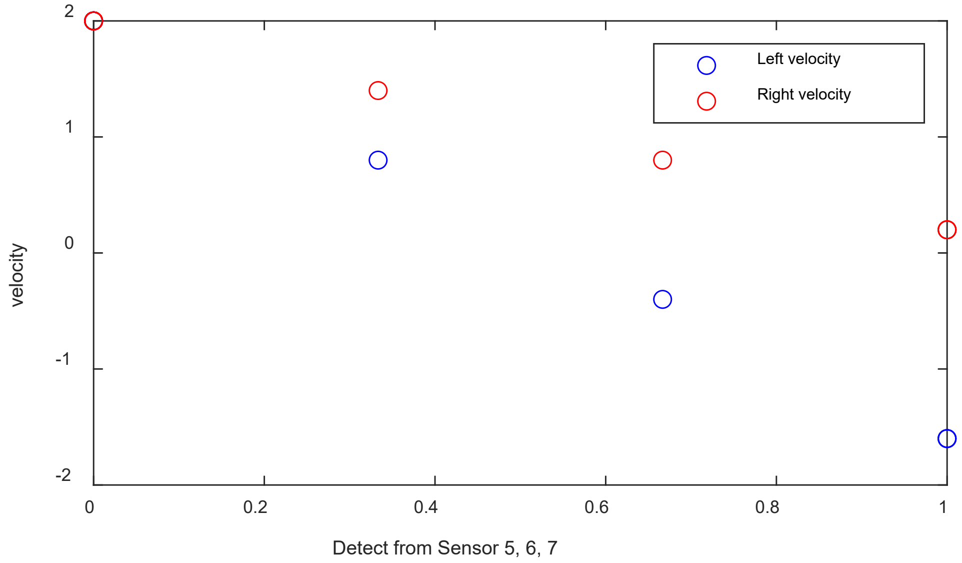
Obstacle at left direction





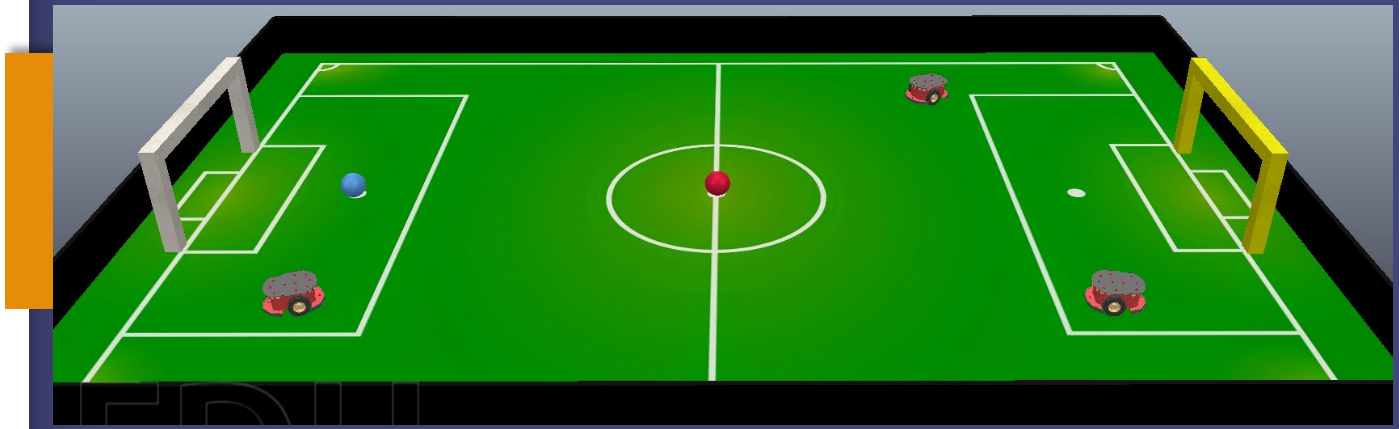
Obstacle at right direction





ASSIGNMENT 1

1. Download and install the robot simulator software CoppeliaSim.
2. Open the provided soccer robot simulation (see the figure).
3. Run the simulation with start button.
4. Learn the programming script that perform the robot motion.



THANK YOU