Hochschule Esslingen
University of Applied Sciences

Campus Göppingen

Obstacle avoidance for Unitree Go2

Finn Schröter, Yannick Tetzlaff, Jonas Borkowski, Laurin Hoyler, Prof. Dr. rer. nat. Johannes Baumgartl – SS 2025

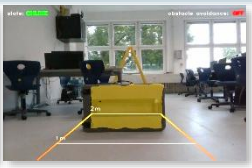
Summary

The project enhances the Unitree Go2 robot dog with autonomous capabilities. Using 3D-printed mounts, we attached a power bank, a NVIDIA Jetson computer, and an Intel RealSense depth camera to the robot. A Python program, running on the computer reads controller input via Bluetooth and sends the commands via WiFi to the robot's firmware which is then responsible for controlling the motors to perform various movements. The depth camera provides visual distance data, which the Python program streams to an HTML client via WiFi as well. Based on this distance data, the robot is able to detect obstacles and adjust its movement for autonomous collision avoidance.

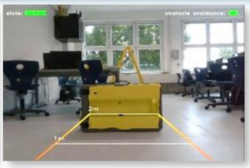
The Project

Image processing and assistance modes

Distance
The colors in the live footage show where the dog might collide, changing colors as obstacles get closer.



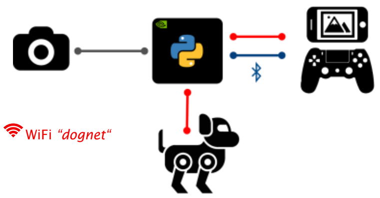
Guidelines
The lines projected on the floor help visualize the dog's size and estimate the position of nearby objects.



Obstacle Avoidance
The dog side steps obstacles and stops automatically if a collision is unavoidable. Status is shown in the GUI.

The program receives an RGB image, and a depth point cloud from the camera. It applies a 60×40×200 cm sized 3D mask to the point cloud and generates a distance-based color-map. Furthermore, it adds an overlay with guidelines and system statuses and combines all into the final image.

Network structure



Setup:
- Depth camera (Intel RealSense D435i)
- Robot dog (Unitree Go 2)
- Computer (NVIDIA Jetson)
- WiFi access point router
- Controller (Dualshock 4)
- Smartphone

The computer is running a Python program. It reads and processes data from the camera (wire connection) and the controller (Bluetooth connection, "pygame" library). The dog operates in a WiFi network called *"dognet"*. The Python program sends controls to the dog's firmware via WebRTC. The video stream connection between computer and smartphone also runs via this WiFi. As the WiFi router is located on the dog and supplied with power from the power bank, the dog is able to be controlled outdoors.

Control

The assistance modes can be turned on and off individually

Emotes:
- Sit down
- Stand up
- Lie down
- Wave

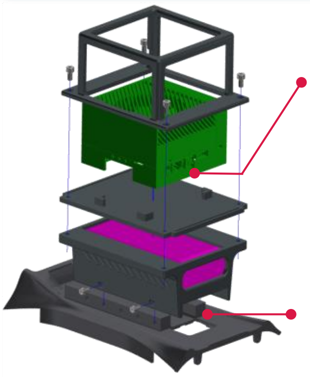
The joysticks control the movement and speed of the dog:

Left Stick: forward/ backward/ left/ right
Right Stick: rotate in place

Collision avoidance

- The collision-avoiding system compares the depth values on the left and the right side of the image.
- If the depth values on one side are smaller, the system interprets that as a nearby obstacle on this side and steers towards the other (see picture).
- If there are the same number of equal depth values on both sides the system interprets this as unavoidable obstacle and stops.

3D Model



The dog's backpack securely holds the power bank and computer. The top section is reinforced to protect the computer from impacts, and the middle plate includes small brackets for cable management using zip ties.

The backplate of the dog was redesigned and equipped with a mounting system featuring multiple holes, allowing future groups to easily attach their own modules.

www.hs-esslingen.de/gp