Design of Efficient Software Tools for the Control of Life-sized Humanoid Robots
Calgary Dynamics | 500-GE2

Logan Jones, Ryan Huynh, Nour Jaffan, Francis Nacu, Aryan Taneja, Toshi Biswas
Schulich School of Engineering, University of Calgary

ROBOTARIUM

SCHULICH School of Engineering

UNIVERSITY OF CALGARY

Abstract

Our project is dedicated to enhancing a humanoid robot tailored for safe and efficient operations in hazardous environments. We're tackling several critical aspects to bolster its capabilities.

- Firstly, we're implementing an advanced 3D mapping system utilizing LiDAR technology. This system enables post-op environmental analysis by creating detailed maps of the area the robot is deployed.
- Secondly, we're developing algorithms to assess the robot's stability dynamically. By analyzing points of contact and center of mass, the robot can determine its stability status and alert operators and its internal systems to potential risks of instability, thereby minimizing accidents and ensuring safer operations.

Introduction

- Welcome to our project! We're enhancing a humanoid robot designed for hazardous environments. Picture a robot that can venture into dangerous areas, such as disaster zones or industrial sites, where sending humans would be too risky. That's the vision driving our work.
- Imagine a scenario where operators can analyze the environment the robot was deployed with a detailed 3D map, enabled by advanced LiDAR technology. This should assist in scenarios such as; structural analysis or mapping safe pathways to victims of a collapsed building for first responders.
- Stability is crucial in dangerous situations. That's why we're developing a feature that helps the robot stay balanced in real-time.

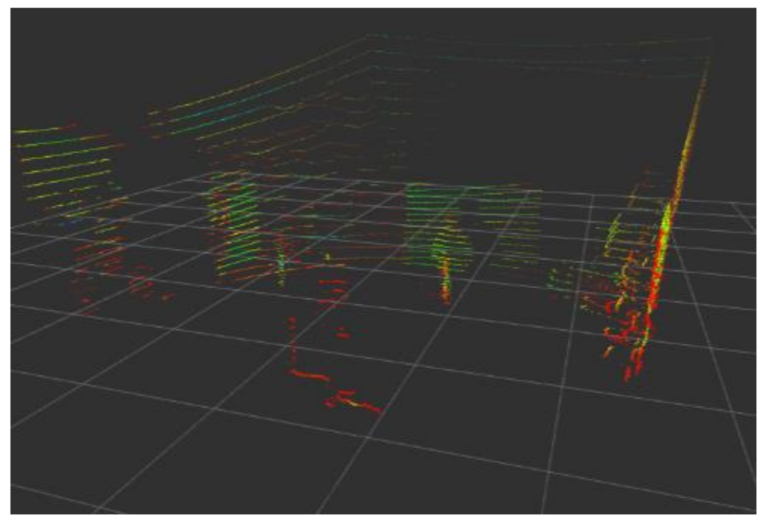
Discussion

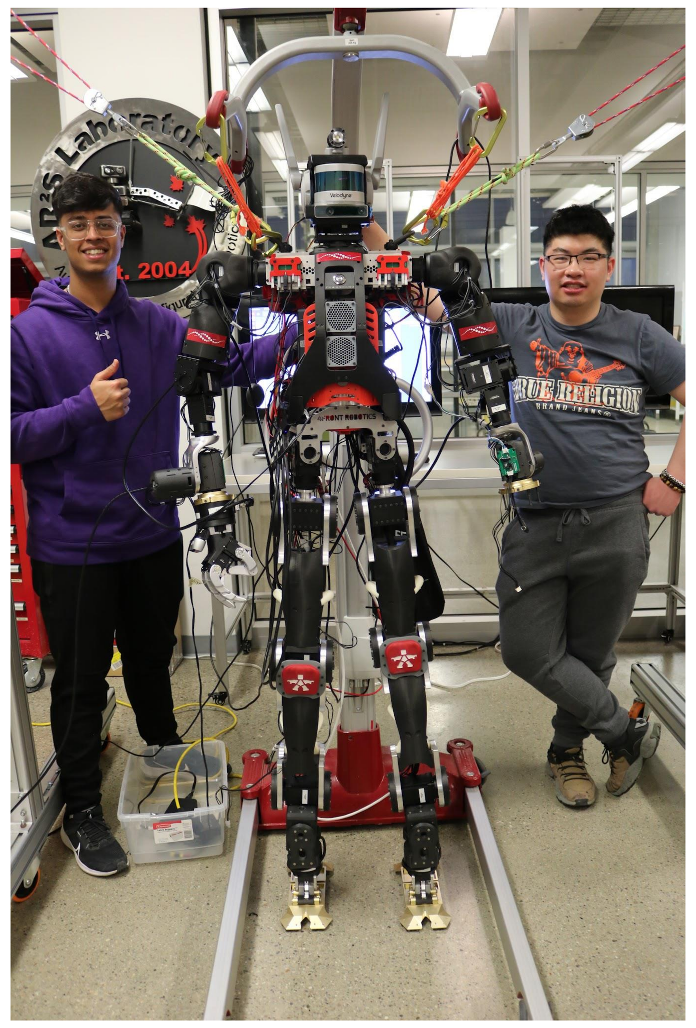
<u>Our project is meant to fulfill these ideas:</u>

Improved Safety: Our project makes hazardous environments safer by giving the robot better senses and balance, reducing the risk for human workers.
Less Risk for Humans: With our upgrades, humans don't have to do dangerous tasks anymore. The robot takes on the risk, keeping people safe.
Easier Remote Control: Our project lets operators control the robot from a distance, so they can stay safe while getting the job done in dangerous places.

Conclusions

In conclusion, our project represents a significant step forward in enhancing humanoid robots for hazardous environments, focusing on improving safety, reducing human risk, and facilitating remote operations. Through advancements in perception and stability assessment, we've made valuable contributions to the field of robotics, with practical implications for industries such as search and rescue, disaster response, and industrial inspection.





Results

Addressing Orientation Challenges: In addressing orientation comprehension challenges, we gained insights into the complexities of sensor fusion and filtering techniques. We learned the importance of implementing sophisticated algorithms to combine data from multiple sensors effectively, enhancing the robot's ability to perceive its orientation accurately.

Collaborative Problem-Solving: Collaboration played a crucial role in overcoming challenges, as we leveraged the diverse expertise within our team to brainstorm solutions and troubleshoot issues effectively. We learned the importance of open communication and teamwork in navigating complex technical problems.

Methods and Materials

In our project, we leverage the Robot Operating System (ROS) as the foundational framework for controlling and coordinating the actions of the humanoid robot. ROS provides a robust platform for developing complex robotic systems.

ROS

Python is our go-to language for most of the coding work in the project. It's known for its simplicity and readability, making it easy for us to write and understand the code. We use Python to develop the logic that dictates the robot's behavior, such as algorithms for mapping its surroundings, assessing stability, and interacting with operators. Python also integrates seamlessly with ROS, allowing us to control the robot's actions efficiently.

While Python is great for many tasks, sometimes we need more speed and efficiency, especially for performance-critical operations. That's where C++ comes into play. We use C++ for tasks that require fast processing, like analyzing large amounts of sensor data or controlling hardware components in real-time. C++ gives us greater control over low-level system interactions and ensures optimal performance for demanding tasks.

References

1. ROS.org. "Robot Operating System (ROS)." Retrieved from https://www.ros.org/
2. Ramirez-Serrano, A. Director, UVS Robotarium Robotics Lab, Department of Mechanical Engineering, University of Calgary. https://www.uvs-robotarium-lab.ca/

CONTACT
Logan Jones (Project Manager)
Email: logan.jones1@ucalgary.ca
Phone: (825)994-1581