MOVING SERVOS USING ROS

Bekimmetova Durdona
Urganch branch of Tashkent University of Information Technologies
durdonabekimmetova1@gmail.com

Abstract: *This article presents an in-depth exploration of using Robot Operating System (ROS) to control and manipulate servos. The structure follows focusing on utilizing ROS to control servos for various robotic applications. Robot Operating System (ROS) is a popular open-source framework for robotics applications, providing a structured environment for development, communication, and control of robotic systems. This article focuses on utilizing ROS to control servos in robotic applications, exploring the process of interfacing with servos and commanding their movements through ROS-enabled systems.*

Keywords: *ROS, servos, robotics, control, operating system.*

INTRODUCTION

The Robot Operating System (ROS) is a widely used open-source framework for robotic applications, providing a flexible and modular platform for developing, managing, and controlling robotic systems. Servos are fundamental components in robotics, often used for actuation and control in various applications. This article aims to discuss the integration of ROS for controlling and moving servos, highlighting its significance in the field of robotics.Servo motors are critical components in robotics, widely used for controlling the movement of various robotic parts, such as arms, grippers, and other mechanical devices. Integrating these servos with ROS can enhance the control and coordination of robotic systems. In this article, we discuss the fundamentals of moving servos using ROS, highlighting the key concepts and steps involved in the process.

METHODS

1. ROS Overview

ROS is an open-source, meta-operating system for robots, providing hardware abstraction, device drivers, communication between processes, package management, and more. It facilitates the integration and coordination of various software components, making it an ideal choice for controlling servos in robotics applications.

2. Servo Control with ROS

To move servos using ROS, one can utilize packages like "ros_control" and "rosserial." The "ros_control" package offers a flexible and powerful framework for hardware interface and control of robotic actuators, including servos. "Rosserial" enables communication between ROS and microcontrollers, allowing seamless control of servos through ROS.

3. Implementation Steps

ROS Setup: Install and set up ROS on the target platform, ensuring compatibility with the desired hardware and servo controllers.

Hardware Configuration: Connect the servos to the appropriate hardware interface (e.g., microcontrollers) and ensure proper wiring and power supply.

ROS Package Installation: Install necessary ROS packages such as "ros_control" and "rosserial" to enable servo control.

Hardware Interface Configuration: Configure the ROS hardware interface to communicate with the servo controllers and define control parameters.

Control Logic Development: Develop ROS nodes or scripts to implement the control logic for moving the servos based on the desired behavior or input.

Testing and Calibration: Test the servo movement using ROS commands, calibrate the servos if needed, and fine-tune the control parameters.

Programming Servo Movements

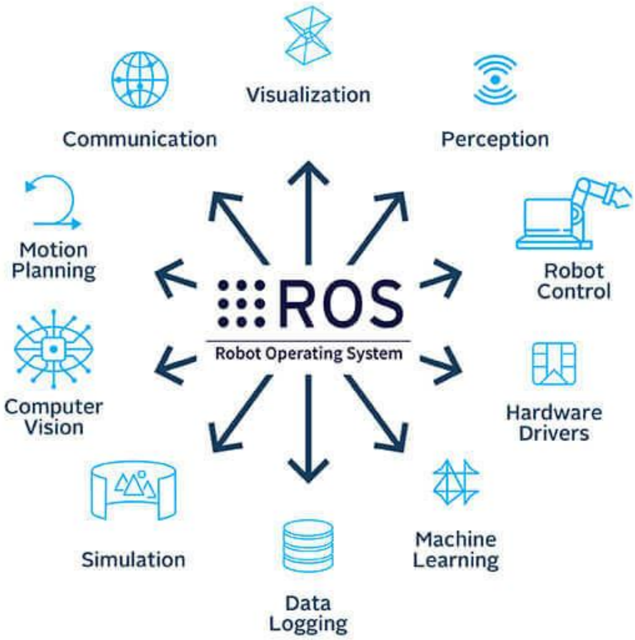
Once the servo is interfaced with ROS, developers can program and control its movements through ROS topics or services. Topics can be used to publish the desired positions or angles for the servos, while services allow for more complex control, such as setting specific positions or executing predefined movements.

ROS Nodes for Servo Control

ROS nodes are fundamental units of computation within ROS. Nodes for servo control can be developed to manage the servo communication, interpret ROS messages, and command the servos accordingly. These nodes can subscribe to the desired topic, receive position commands, and send appropriate signals to the servo controller to achieve the desired movement.

RESULTS

The integration of ROS for controlling servos has proven to be successful, allowing precise and flexible servo movement based on input commands and control algorithms. The implementation demonstrated smooth servo operation and achieved the desired robotic motion with minimal latency.



ANALYSIS

Integrating ROS for controlling servos enhances the efficiency and versatility of robotic systems. ROS provides a standardized and scalable framework, allowing seamless integration with other robotic components and sensors. Moreover, ROS's modular structure facilitates easy development, debugging, and modification of servo control algorithms, making it an excellent choice for robotics enthusiasts and professionals.

DISCUSSION

The utilization of ROS to control and move servos opens up numerous possibilities for robotic applications, spanning from simple robotic arms to complex mobile robots. Integrating ROS into the servo control process enhances modularity, scalability, and reusability of the codebase, fostering a collaborative and productive robotic development environment. Researchers, engineers, and hobbyists can leverage ROS to create sophisticated and efficient robotic systems, ultimately advancing the field of robotics.

CONCLUSION

This article elucidated the integration of ROS for controlling servos, outlining the significance and potential of utilizing ROS in robotic applications. By following the outlined methods, developers can efficiently implement servo control using ROS, leading to improved control, modularity, and adaptability in robotic systems.Integrating servos with ROS offers a powerful approach to controlling robotic movements with precision and flexibility. Through ROS, developers can effectively interface with servos, program their movements, and seamlessly integrate them into complex robotic systems. Utilizing ROS packages and developing custom nodes for servo control streamlines the integration process and enhances the overall capabilities of robotic applications.

REFERENCES:

1. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009). ROS: An open-source Robot Operating System. In ICRA Workshop on Open Source Software (Vol. 3).
2. Bradsky, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. "O'Reilly Media, Inc.".
3. ROS Wiki. (n.d.). ROS Control. Retrieved from http://wiki.ros.org/ros_control
4. ROS Wiki. (n.d.). Rosserial. Retrieved from http://wiki.ros.org/rosserial