Chelsea Lau
EC601 Product Design in ECE
9/18/2022

PROJECT 1: TOPIC IN ASSISTIVE ROBOTICS

As assistive robotics, and robotics in general, exist at the intersection of multiple disciplines, mechanical, electrical, and software, there are a multitude of aspects one could dive deeper into. Some examples include the dynamics and vibrations of robotic manipulators, sensing and perception, or control systems. Another, would be the operating systems used on these robots.

What is an operating system? For every computer, there needs to be software that handles tasks such as scheduling tasks, memory management, input and output, and managing peripheral devices. The most commonly known operating systems are Microsoft Windows, Apple's macOS and iOS, and Google's Android OS. Though robots do have computer systems onboard, they often employ different operating systems, one of the most popular being an open-source meta-operating system named ROS (Robot Operating System). Formally it is not an operating system because it is implemented as a framework on top of Linux Ubuntu, but it does provide some of the same services including hardware abstraction, low-level device control, and message-passing between processes. It also allows and encourages developers to reuse code from projects executed across the robotics research community and industry.

To give an overview of how ROS actually works, the framework can be simplified into a "core," which is middleware that contains tools for communication, and libraries. The middleware allows developers to divide up the functionality of their applications into smaller programs, since it manages the communication between them. These programs are known as "nodes" in ROS and every node manages one task. The communication tools included in the middleware are topics, services, and actions. Topics are the channels through which nodes can communicate by passing messages. Services provide an alternative form of communication through a synchronous client/server system. Services consist of pairs of messages, requests and replies. Actions on the other hand provide asynchronous client/server communication and are based on topics. They allow clients to send long-term requests, monitor the status of the server asynchronously, and cancel their requests at will. As for encouraging code reusability, developers can create packages to be used by others for future projects.

One such package called gt-ros-pkg was developed by Georgia Tech and contains ROS code from multiple robotics laboratories: the Healthcare Robotics Lab, the Socially Intelligent Machines Lab, the Humanoid Robotics Lab, and the Human-Automation Systems Lab. The Healthcare Robotics Lab in particular has used ROS for two of their assistive robotic systems, EL-E and Cody.

Cody is a robotic nurse that was developed to autonomously perform nursing tasks, and has been designed to execute these tasks with minimized force of impact and pressure with elderly and disabled patients in mind. One of its main capabilities is assisting patients with limited mobility with bathing in the form of bed baths. Another is opening doors, drawers and cabinets. In order to accomplish these tasks, Cody uses code packaged into the 2009_humanoids_epc_pull package, which allows it to use equilibrium point controllers. By utilizing the ROS framework to develop code that can control the motion of a robotic arm through adjusting the position of a Cartesian-space equilibrium point, the developers at the Healthcare Robotics Lab have minimized the efforts of future researchers hoping to employ equilibrium point control on their own systems.

Another notable ROS package created by the Healthcare Robotics Lab is assistive_teleop. The code provides a web browser interface to use with the Personal Robot 2, from Willow Garage, specifically designed for users with severe motor limitations. It is to be used in conjunction with an assistive HCI (Human-Computer Interaction) device, which is a mouse or gamepad that allows users to access computers hands-free. Instead, users can move another body part, usually the head, to control. The interface's features include low-level control of all DOF of the Personal Robot 2, visual feedback, and semi-autonomous functions that can manipulate the system to assist in completing Activities of Daily Living (ADLS).

Evidently, engineers in assistive robotics research, and undoubtedly industry as well, heavily employ ROS for their robotics development. The framework provides key functionalities of an operating system while simultaneously increasing the efficiency of development by offering a robust system for code reuse. As such, ROS will continue to be instrumental in the advancement of assistive robotics.

References:

https://www.tutorialspoint.com/operating_system/os_overview.htm
http://wiki.ros.org/ROS/Introduction
https://www.allaboutcircuits.com/technical-articles/an-introduction-to-robot-operating-system-ros/
https://coe.gatech.edu/news/2010/11/robotic-cody-learns-bathe
https://www.ros.org/news/2010/03/robots-using-ros-georgia-techs-assistive-robots.html
https://wiki.ros.org/gt-ros-pkg
https://wiki.ros.org/2009_humanoids_epc_pull
https://www.robaid.com/robotics/cody-mobile-manipulating-robot-could-aid-us-in-future.htm
https://wiki.ros.org/assistive_teleop
https://roboticsbackend.com/what-is-a-ros-service/
https://roboticsbackend.com/what-is-ros/
https://smartech.gatech.edu/bitstream/handle/1853/37359/humanoids09_epc.pdf