Hardware Accelerated Mobile Robot Manipulator Platform

Project Proposal

School of Electrical and Computer Engineering

Advisor: Professor Bruce Land

Team: Boxuan Ai/ba394, Kowin Shi/kss223, Tian Qiu/tq42, Tiange Zhao/tz377

Team Number: 18

Project Advisor Signature: ______________________

ABSTRACT

This project aims to create a mobile robot manipulator platform from scratch, and to develop all custom hardware and software designs. To achieve this goal, a team of ECE, MAE and CS MEng students have been built to accomplish all of the hardware and software engineering, as well as manufacturing. The team is also coordinating research and reaching out to professors who are proficient in their fields, for challenges that are deemed too difficult for the team's abilities. The final deliverable is a low-cost mobile robot that has a 6 DOF arm, capable of high speed SLAM, as well as vision-based navigation and pick-and-place. This is significant because there has not been open-sourced designs of similar scale, and robotics enthusiasts can build from our design to have their own manipulator platform for a fraction of the cost of the Kuka Youbot ($30k).

INTRODUCTION

The central aim of this MEng project is to create a custom 4-wheel-drive mobile robot platform that is open-source, relatively affordable, compact in size, and have enough compute horsepower for high speed SLAM and path planning for both the mobile robot and the manipulator arm. The motivation for this project is that there is not a similar platform existing on the market except for the poorly integrated Turtlebot series. The robot will have 2 quick release docks for attachments at either end, to fit the manipulator arm and a miniature cyclone-dust-separating vacuum module. It will rely on ROS for its software framework, and there will be plenty of custom code to interface with the various hardware systems onboard.

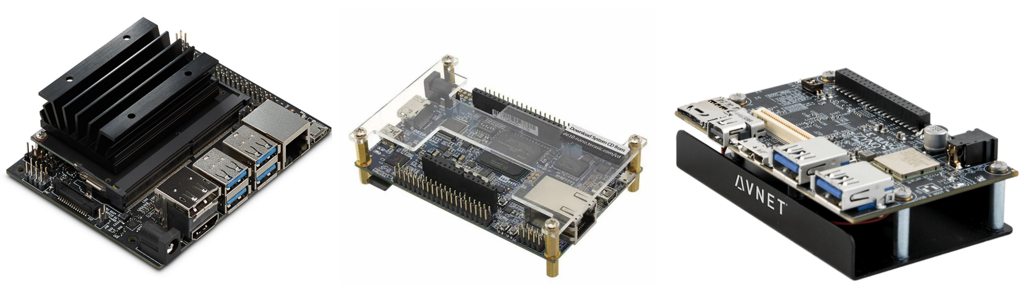

Figure 1. Compute Hardware Boards (Images Courtesy of Manufacturers' Websites)

The robot will carry an Nvidia Jetson Nano for CUDA acceleration, an Intel/Altera DE10-Nano for FPGA acceleration, and a Xilinx Ultra96 (Zynq Ultrascale+) board for additional FPGA processing power. The purpose and function of each board is elaborated below.

ISSUES AND CHALLENGES

Figure 2. Mechanical CAD Design and LEGO Bricks (Image Courtesy of Google)

The final challenges that the robot aims to solve are: 1. High speed robot vacuum with automatic emptying via manipulator arm and 2. Vision based pick and place objects with manipulator arm (cleaning up a pile of legos). A potential third high-level functionality could be Professor Skovira's UV cleaning project, depending on time. The detailed technical breakdown of the challenges into subproblems is discussed below.

APPROACH

Hardware:

Almost everything will be designed and made from scratch, from the mobile robot to the 6-DOF manipulator arm to the vacuum module. The arm has already been designed and manufactured:

Figure 3. Completed Hardware, Arm and Robot Platform

The vacuum module is still in the research phase, and the following projects are being referenced: DIYson, Dual Stage Cyclone.

The mobile robot base is currently being designed, and manufacturing has already been mostly completed. It is driven by 4x NEMA17 stepper motors (for more accurate control inputs) with a choice of TMC5160 or DRV8825 high current drivers, and powered by 20x 18650 lithium-ion cells (4s5p) salvaged from a Tesla battery module. The goal is to make it autonomously rechargeable, with it being able to return and mate with a wireless docking station when power is low.

In terms of the drivetrain, the team intends on exploring both holonomic and nonholonomic drives as the 4-wheel-drive capability will allow for either. This will be very interesting to see in terms of the differences in control algorithms.

In terms of sensors, the team intends on using the [RPLIDAR A1M8] lidar for mapping and navigation, and either an intel realsense camera or just a raspberry pi camera for computer vision. Bump sensors may be added depending on complexity/necessity.

Software:

The current plan is to use the DE10-Nano for mapping/navigation, as it is able to leverage its FPGA for massive parallelization of particle filter in the FastSLAM algorithm, and process the raw lidar data with as little delay as possible (lidar is max 8khz sampling rate). This results in a ultra-high update frequency for better localization performance, and thus allows for higher velocities in its trajectory. This requires Verilog/System Verilog designs on the FPGA, and C++ interface on the HPS side for the lidar interface. The team is leaning towards using ROS to simplify the SLAM map storage/visualization functionalities, although finding an appropriate linux OS may require some research (since it has to work with the ARM SoC).

With the safety critical computation offloaded to the DE10-Nano, the team is planning to use the Jetson Nano as the high-level planner for the whole robot to issue commands and dictate trajectories, as well as perform the less critical controls of the 6-DOF manipulator. The manipulator computation may get offloaded to the Ultra96 board, if it becomes too computationally taxing. The Jetson Nano will most certainly be running ROS, so some combination of python/c++. Some of the vision components may also be run on the Jetson, as it has 128 Maxwell GPU cores and Nvidia specifically advertised it for vision/robotics applications.

The Xilinx board will be used for further hardware acceleration, as it is has one of Xilinx's newest Zynq FPGA processors combined with a quad-core HPS. In particular, it will be used for neural net acceleration in object detection algorithms. The FPGA is large enough to also run parallelizable path planning algorithms for the manipulator arm, if deemed necessary.

In terms of inter-computer networking, both the DE10-Nano and the Jetson Nano have ethernet ports, while the ultra96 supports ethernet through usb. This part is still to be determined upon more discussions with Professor Land, but most likely all 3 will communicate with UDP protocol through a network switch, and either the ultra96 (which has wifi built into the pcb) or the Jetson Nano (has wifi M.2 expander card) will handle external communications such as to another computer for visualization and GUI.

Specific algorithms that the project aims to implement are: FastSLAM, RRT/Visibility Roadmaps, Bug, Nonlinear MPC, Joint Forward/Inverse Kinematics, RCNN.

Work Delegation and Timeline:

Due to the complexity and size of this project, The team has compiled individual smaller tasks with specific details on a separate spreadsheet, linked here. At the beginning of the semester, the team members went through the spreadsheet and each member wrote a personal task description that was submitted to Professor Land. Please see them in the appendix below. In short, Boxuan will take on electrical hardware then transition to computer vision, Kowin will take on mechanical and electrical hardware then transition to robotics algorithm design, Tian will take on object avoidance and computer vision, and Tiange will take on embedded software then move onto navigation design.

In terms of project timeline, the team aims to complete hardware prototype by early October, and receive final hardware by early January. For software, the team aims to accomplish lidar integration with ROS navigation and manual wireless control by December, as well as completing embedded software by then. The team hopes to complete manipulator and mobile robot base path planning by March, and finally computer vision pick and place by April of next year.

SUMMARY

In conclusion, the final purpose of this MEng Design Project is to create a custom wheel-drive mobile robot platform with hardware acceleration. It will be capable of numerous high-level applications based on the platform, and the team specifically aims to achieve high speed SLAM and path planning for vacuuming, and vision-based pick and place for objects. Those challenges are chosen because they are common applications of robotic automation in everyday life, and allows the work of this project to draw direct comparisons with industry.

APPENDIX

Member: Boxuan Ai
Credits: 3

Individual Tasks (and how it contributes to the overall goal):

1. Design 2 PCBs: one for mobile robot base and one for arm
2. Robot base PCB:
    a. Design DC to DC power converters. From battery voltage to computer voltages. Input 10-17VDC, output 5V 6A, 12V 2A, 6-8.4V 6A (last one optional).
    b. Design MCU system to handle stepper drivers, including settings and commands through SPI.
    c. Design wireless charging system for the robot. Make the charging circuit part of the PCB, boost converter 5V to 16.4V.
    d. Measure the voltage of the battery pack and DC power points.
    e. Implement force feedback control on the gripper of the robot arm with ADC (resistive pressure sensors).
3. Implement remote instant control of robot direction and speed control by a simple gamepad.
4. Computer vision aims: be able to read the QR code, in order to figure out the charging position and being charged automatically.

Milestones and Deliverables:

1. PCB prototype demonstration. With stepper drivers, DCDC converters and wireless charging circuits implemented.
    a. Prototype system using breadboard
    b. Prototype system with milled PCB
    c. Final product with professionally made PCB
2. Robot arm gripper control: be able to grip some determined object.
3. Robot moving control by joystick in gamepad.
4. Computer vision realization.

Member: Kowin Shi
Credits: 4

Individual Tasks (and how it contributes to the overall goal):

1. Manage project, make sure members' questions are answered and everyone is making progress.
2. Mechanical design and fabrication of both the mobile robot base and the 6 DOF arm.
3. Simulation setup of both the robot base and robot arm in Gazebo and Rviz.
4. Develop optimal control policy for the mobile robot base to follow trajectories.
5. After initial Lidar prototype software is complete, move SDK to DE10-Nano and developed custom ROS node for parallelized SLAM.
6. Develop vacuum algorithm (sweep and fill) and path planning to execution routine.

Milestones and Deliverables:

2. Complete both mobile robot base and arm, fix any issues that may arise in testing
3. Simulator setup
    a. Simplify robot models in CAD, export mesh and URDF files
    b. Familiarize with Gazebo, figure out how to import custom robot files
4. Optimal control
    a. After robot moves from high level commands, get access to overhead localization system
    b. Start with simple velocity waypoint following, establish baseline performance
    c. Increase speed until overshoot becomes too much, read papers to figure out dynamics model and develop MPC control policy
5. Parallelizable SLAM
    a. Read papers to figure out parallelizable SLAM and resource requirements
    b. Develop and test FPGA design
    c. Tie in with Lidar readings from HPS
6. Sweep algorithm
    a. Develop algorithm in Gazebo first for easy testing
    b. Experiment with robot in low speed, then increase to see max speed doable

Member: Tian Qiu
Credits: 3

Individual Tasks (and how it contributes to the overall goal):

1. Design robot vision recognition and localization algorithm to process real-time frames to avoid driving downstairs. This makes sure that the mobile robot moves around and completes tasks safely.
2. Developed objection detection and segmentation algorithm based on robot vision to get the shape, color and pose of Lego for further picking up. This completes the high-level software part of cleaning up a pile of legos.
3. Figure out network protocol between Jetson Nano, DE10-Nano and Ultra96. This completes the communication and transfer mechanisms of the platform. Do further testing with demo and benchmark bandwidth. (optional)

Milestones and Deliverables:

1. For vision algorithm to avoid drive downstairs:
    - Build web crawler scripts to download images from websites and preprocess raw images to generate valid input data
    - Collect at least 1,000 staircase images, preferring to be taken from downward as the train set, and another 100-200 images as the test set
    - Read related paper and train neural network with CUDA in PyTorch framework on Linux
    - The final algorithm aimed to achieve over 60% accuracy when recognizing and localizing staircases from downward and less than 1-minute cost per frame in inference.
2. For objection detection algorithm:
    - Collect at least 2,000 lego images, preferring to be taken from different directions as the train set, and another 500 images as the test set (Maybe the number of images taken from the commonest directions should be most)
    - Python programming and train neural network with CUDA in PyTorch framework on Linux
    - Applied well-trained neural network model to robot platform and test its performance
    - The final algorithm aimed to achieve over 80% precision when classifying legos in terms of their shapes and color and over 60% in detecting their orientation.

Member: Tiange Zhao
Credits: 3

Individual Tasks (and how it contributes to the overall goal):

1. Develop the ROS nodes to control robot arm from Lewansoul LX-16a servos. Get information about the servos including joint states, servo speed, then calculate the posture of the arm.
2. Learn basic structure of Slamtec RPLIDAR Public SDK. Use it to get data from the lidar based on ROS.
3. Help develop the computer vision algorithm, achieve the Lego objects recognition and localization function using the data from lidar.

Milestones and Deliverables:

1. For robot arm control:
    - Build the basic data collection node to get the real-time information from the servos through serial port and send control messages to the servos.
    - Servo performance characterization:
        - Determine control/polling frequency, accuracy, and control parameters.
        - Experiment with position control using only velocity commands, if feedback loop if fast enough, and record response data.
    - Using a remote controller to control the posture of the robot arm manually.
    - Calculate the posture matrix of the robot arm and the location coordinates of target objects.
    - Final control system aims to make the arm automatically reaching a certain point in the coordinate system.
2. For lidar data collection and computer vision algorithm:
    - Build the data collection node by using SDK to get the real-time information from the lidar.
    - Process lidar data and calculate the coordinates of target object in robot arm coordinate system.
    - Combine object detection algorithm with arm control system. Achieve that the arm can reach to the location of the target object automatically.
    - If the gripper design is accomplished, add the gripper control to the system and achieve Lego objects detection and capture.