ENEE 408I Capstone Design Course in Autonomous Robotics

Spring 2018 Project

Autonomous Robots, Deep Learning, and Wearable Sensors in Rehabilitation[1]

G. Blankenship and J. Renner

1 Introduction

This semester in the ENEE408I Capstone Design Course in Autonomous Robotics we will design and implement autonomous robots with onboard smart devices and sensor systems to support the rehabilitation of disabled or elderly persons.

1.1 Support Robots

Specifically, we will consider several problems related to robots supporting people in their homes:

- **Monitoring:** Design, implement, and test an autonomous mobile robot for person monitoring using a camera. The challenge is to assess the relative health of a subject going about their daily lives based on camera images captured by the robot. The robot processes the images using an onboard computer or remote server and makes a determination about the health of the subject. If the robot detects a potential problem, it issues an alert to a remote server or an alarm to an authorized person or service. In any case the robot provides regular reports on the health of its subject.
- **Companion:** Design, implement, and test an autonomous mobile robot as a "companion" to a person. The robot provides an interactive service, using the Amazon Echo device to solve small problems for the person (turn on/off the lights, TV, etc.), monitors the environment (is the stove on?), and monitors the health of the subject (amount of movement, sleep, etc.). The robot also provides entertainment services and information interaction, similar to Amazon Echo device. The robot provides reports to a remote server on the status of its subject and the extent of the interactions.
- **Optional wearable sensors:** Design, implement, and test an autonomous mobile robot for person monitoring using wearable sensors. The challenge is to assess the relative health of the person going about their daily lives based on sensors worn by the subject. The robot receives the sensor data, processes it, and makes a determination about the health of the subject. If the robot detects a potential problem,

---

[1] This course is sponsored in part by gifts from BAE Systems and Boeing.

it issues an alert to a remote server. In any case the robot provides regular reports on the health of its subject.

- **Student defined project** for a robot to support health monitoring, emergency alerting (falls), etc. The problem and its technical scope must be defined in the first 3 weeks of the semester; and the instructor must approve the problem. Innovation is encouraged.

1.2 Specific Challenges

Each lab session will have 7 teams of 3 students. The teams must design and program their robots to address one or more of the problems summarized above. To be successful, each team will have to address several specific challenges:

1. **Safe interaction:** Since the robots are relatively small, it is important that they be able to avoid being "under foot," causing their subject, or any person in the area to trip and fall. The robot must maintain a safe distance from each person in the area, while remaining close enough to its subject to perform its duties. *This is a key requirement, and its solution is an important part of the course.*
2. **Subject awareness:** The robot must recognize and interact with a specific subject. It must be able to recognize and respond to that person in the presence of several people. This might be done based on sensor signals, image recognition, or an auxiliary signaling device. *Each team will have to produce a solution to this key problem.*
3. **Image and signal processing:** The robots' assessments of their subjects will be based on either images acquired by their onboard cameras and/or signals acquired from the sensors worn by the subjects. It will be very challenging to assess a subject's "health" based on these kinds of signals. *This is likely to be the key challenge of the project.*
4. **Interactions with clinicians:** Each team will have an opportunity to engage with one or more clinicians who are familiar with conditions of the types of subjects the robots are designed to serve. It will be important for each team to provide focused questions and demonstrations to maximize the benefit of the interactions.

The robot designs will be tested in a series of demonstrations. The schedule for these is as follows:

1. **Tuesday February 13, Controlled Motion.** This will be the first test. Each team's robot will have to travel through the lab avoiding obstacles and people.
2. **Tuesday February 27, Person following.** Each team's robot must follow a person walking in the laboratory and hallway outside the lab.
3. **Tuesday March 27, Person identification.** Each team's robot must demonstrate the ability to identify a specific person and respond to (voice) commands from that person.
4. **Saturday April 28, Maryland day.** The class will demonstrate their robot's capabilities to identify persons and process voice commands and/or wearable sensor signals to visitors to the campus.
5. **Tuesday May 8 – Last Class, Class Challenge.** The teams in each lab section will demonstrate their robots responding to specific subjects and providing assessments of their health based on image processing, wearable sensors, or both.
6. **Friday May 18.** Final Reports due.

This is an ambitious schedule, and every member of each team will have to contribute to make it happen. Many researchers and companies have addressed the design of "social robots" for the past several years. Your projects are as challenging as many of these research and commercial projects.

2 Organization of the Class

2.1 Class Lectures

The class will have a single lecture/recitation meeting each week (Monday 10AM, ITV 1111). This meeting will be used to introduce material on mobile robots, especially material on human-robot interaction. It will also be used for team status reports. Each team will be expected to research and present two 20 minute talks on topics related to the theme of distributed robotics. We may have guest lecturers during the semester. A schedule for the presentations will be posted later.

2.2 Laboratory Sessions

Each section of the class will meet for 3 hours each week in the BAE Control Systems Laboratory, Kim Building Room 3209. **Jay Renner,** the Engineer in charge of the lab will help you with hardware and software issues. Professor Blankenship will be in the lab most of the time. We will also have two full time GTAs, Helene Nya Chima Nguewou-Hyousse hnguewou@umd.edu and Usman Fiaz fiaz@terpmail.umd.edu.

You are welcome to use the lab during normal business hours by scheduling time with Jay Renner. ***The lab will not be available at nights or on weekends.***

You can take your robots home with you. They are yours for the semester. While we have some spare parts, be careful with voltage levels, since the wrong level can damage the microcontrollers and other components.

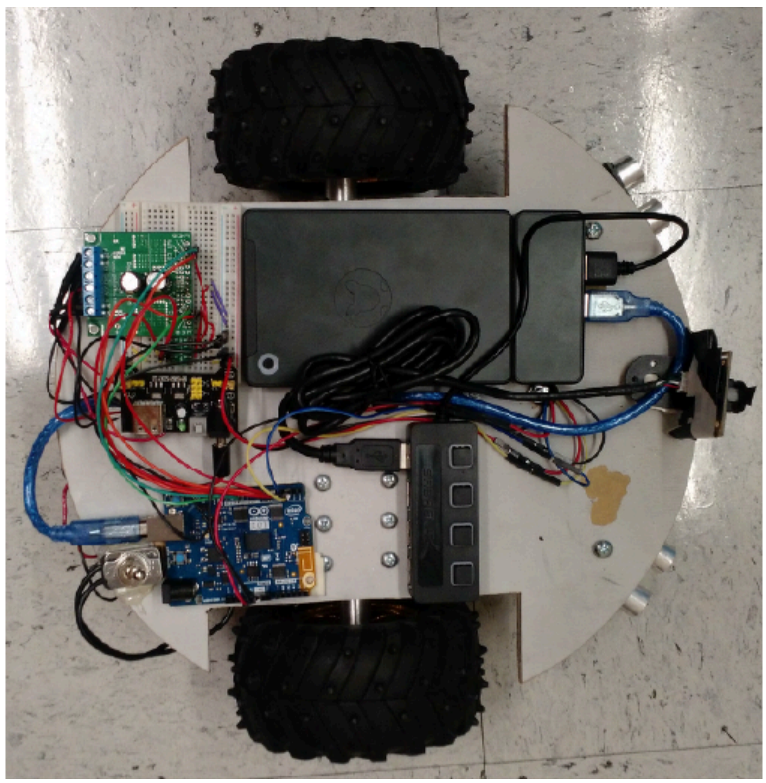

Figure 1 ENEE 408I (2017) robot base showing motor controller (rear top), Arduino for maneuver control (rear bottom), Kangaroo mobile desktop (center, top) for image processing, communications, and control, wide-angle lens camera (front center), and two "Ping)))" sensors for obstacle avoidance (front). The robot will be completely redesigned in 2018.

3 Robot Hardware

In addition to the frame and wheels, the class robot base includes hardware necessary for control, communications, and object detection. An example of the robot base is shown in the figure. The robot has a camera on the front, and a small main computer (this semester we will replace the computer shown in the image) for image and signal processing, subject monitoring, path planning, and communications. It also has an Arduino board for low-level control, onboard sensor signal processing, and serial communication. There are two "Ping" sensors on the platform that measure distances to objects and are used to avoid obstacles.

The Arduino board reads the Ping sensors, and commands the motor controllers, which in turn control the two wheel motors, enabling movement and navigation. The Arduino receives commands from the "main computer" and executes the corresponding maneuvers. It sends sensor data to the main computer for mapping and other tasks. The main computer handles image processing and communications with other robots.

This semester we will use an NVIDIA JETSON TX2[2] as the "main computer" to handle robot-to-robot communications, object detection, and planning tasks. This powerful "micro-supercomputer" is designed for machine learning algorithms, especially deep learning. It gives the class an amazing platform for development and implementation of sophisticated algorithms.

3.1 Amazon Echo for Robot Control and Interaction

In addition to the core components described in the following sections, we plan to add "smart devices," such as the Amazon Echo Dot to the robots to facilitate interaction between the robot and its subjects.

See the following web links for more information:

https://developer.amazon.com/alexa-voice-service

http://smarthome.reviewed.com/features/everything-that-works-with-amazon-echo-alexa

https://www.dexterindustries.com/projects/alexabot-amazon-alexa-controlled-robot/

https://github.com/alexa-pi/AlexaPi

Each robot will have an Amazon Echo Dot that will be used for voice-based interaction with the robot's owner. Each team will have to develop applications that use this technology to assist users of the robot.

3.2 Motors and Motor Controllers

Each robot has two motors and a dual channel motor controller. Each motor has an encoder that provides a measurement of distance travelled. You will need to calibrate this sensor prior to use, and you may find the measurements noisy and unreliable. *We recommend that you not use the wheel encoders.*

[2] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/

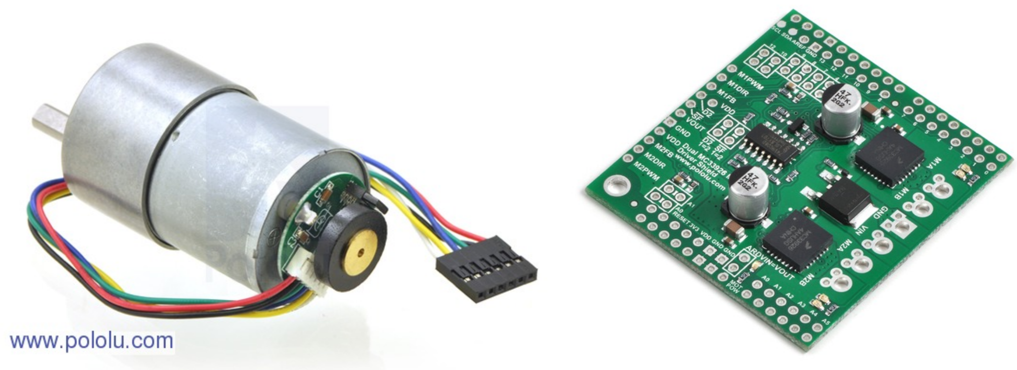

Figure 2 Geared motor with encoder (left) Dual motor controller (right)

Details of the motor and motor controller can be found at the Pololu website:

- http://www.pololu.com/catalog/product/2503
- http://www.pololu.com/catalog/product/1445

3.3 Ping))) Sensors

We will use ultrasound sensors to measure distance over short ranges. Each robot will be equipped with 2 or more Ping))) sensors for obstacle avoidance.

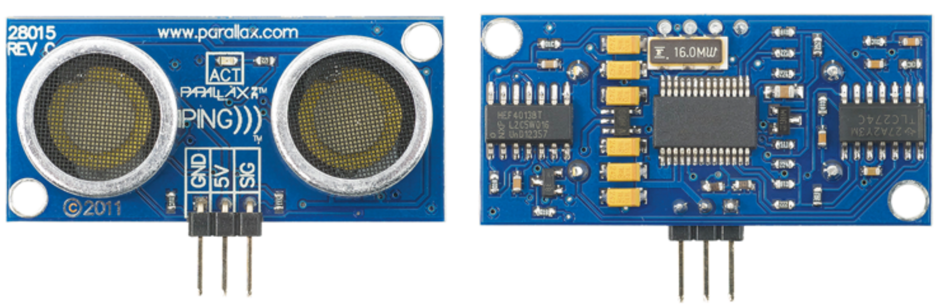

Figure 3 Ultrasound sensor for measuring distance

The sensors have a limited distance and are somewhat noisy. More information on these sensors can be found at the Parallax web site: http://www.parallax.com/product/28015

3.4 Inertial Measurement Unit (IMU)

The IMU provides the sensor signals needed to stabilize the robot and to direct its motion along a path. The unit has a 3-axis accelerometer and a 3-axis gyro. The accelerometer is reasonably reliable; however, you will find that the gyro "drifts" over time. It is not likely to be especially useful for navigation; but you should investigate it.

Figure 4 Inertial Measurement Unit for navigation and control

Data sheets for the IMU and example projects can be found at the Sparkfun website:

https://www.sparkfun.com/products/11028

The Arduino 101 board has a 6-axis accelerometer/gyro; and so, teams using that board will not need a separate IMU.

3.5 Arduino Boards

We will use the Arduino Uno Board or the Arduino 101 Board as the low level controller for the robot. The board will manage maneuver control and the robot's response to the low level (Ping) sensors for obstacle avoidance.

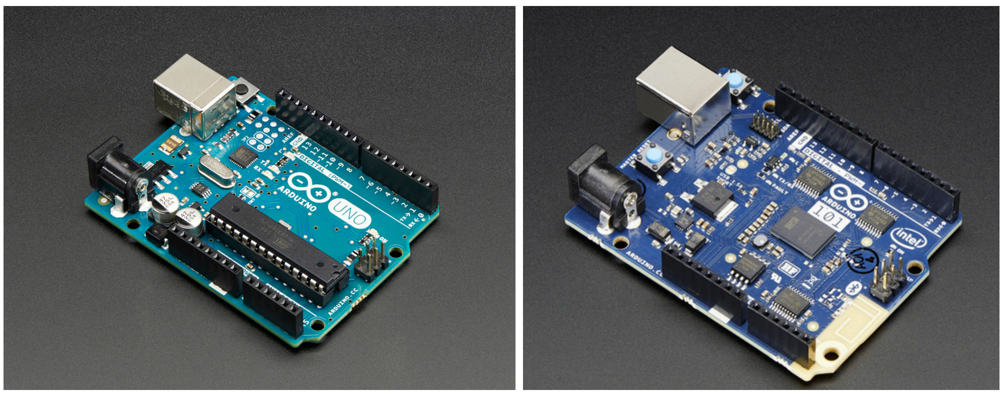

Figure 5 Arduino Uno processor (left) Arduino 101 processor (right)

The main Arduino website is a good place to start. https://www.arduino.cc/ The Arduino software development environment can be downloaded from the site. There are many tutorials – look for the ones using the Ping sensors.

Note: The Arduino Uno runs at 5 volts; **the Arduino 101 runs at 3.7 volts. The Arduino 101 can be damaged if devices expecting 5 volts are plugged into it.** We have had several boards damaged in the past, so be careful with this.

3.6 NVIDIA JETSON TX2 as the Main Computer

The NVIDIA TX2 runs Ubuntu (16.04) and presents a familiar interface when started. The TX2 includes a wireless card, and it can be connected to the UMD network. Ubuntu includes several development tools, especially Python. We will use Python (version 2.7) in our class. We will also use OpenCV for image processing, and it and several other Python libraries need to be installed on the TX2. We will provide a list and installation instructions in the class. The TX2 has built in BLE and WiFi, both useful for our class. We will use the lab monitors, keyboards, and mice to interface to the computer.

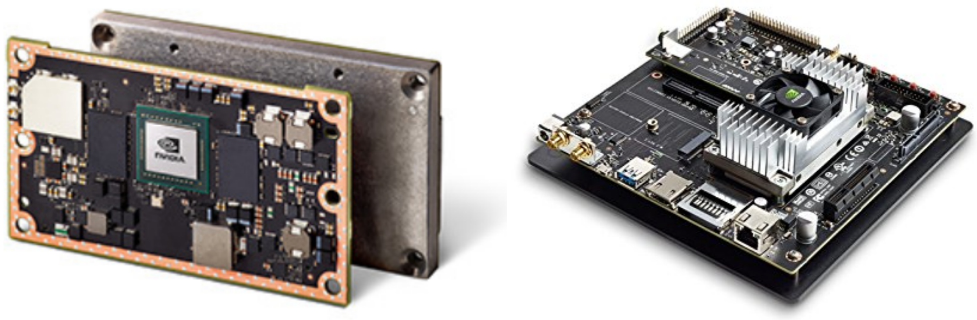

Figure 6 NVIDIA TX2 module with heat sink (left) and evaluation/development kit (right)

Teams may wish to use Team Viewer or VNC software to interface with the JETSON TX2 from their laptops. See https://www.teamviewer.us/

3.7 Raspberry PI as Additional Computer

We also have Raspberry PI computers in the lab for use in the project. They are more powerful than the Arduino, and could be used instead of the Arduino for low level (maneuver) control of the robot, including image processing for maneuver control. The PI includes WiFi and Bluetooth radios, and can easily be connected to the JETSON TX2.

4 Robot Software

4.1 Arduino

The Arduino board will be programmed using the Arduino integrated development environment (IDE). Information on the IDE and the software can be found at the Arduino website: http://arduino.cc/en/Main/Software

Figure 7 Arduino IDE with one of the example programs

The code in the figure shows some of the syntax of the language used to program the Arduino board. Its syntax is similar to *C* and is based on a language called *Wiring*, an open-source programming framework for microcontrollers. The Arduino IDE is based on *Processing* an open source programming environment for creating images and animations.

Information on the Arduino platform along with links to a very wide array of programming examples and tutorials can be found at the website: http://www.arduino.cc/

Arduino code for obstacle avoidance control of a robot based on Ping))) sensors can be found on this web page:

http://learn.parallax.com/project/shield-bot-roaming-ping/let-it-roam

4.2 High Level Controller Software

A key part of the project is to provide a "high level controller" for the robot. The goal is to make it completely autonomous and to enable collaboration among some or all of the bots. There are many possibilities. Since we have chosen the NVIDIA JETSON TX2 as our main computer, we have several choices for the high-level language. This semester we will use **Python** as the language for high-level programming. Python is installed on the TX2 computer. (Both Python 2.7 and 3.0 are installed. We will use Python 2.7.)

There are many useful packages such as SciPy, NumPy, matplotlib, … (See https://pythontips.com/2013/07/30/20-python-libraries-you-cant-live-without/). They can be installed on your system using commands at the terminal prompt.

The site https://www.python.org/ has comprehensive information on Python and its applications. The Python tutorial here https://docs.python.org/3/tutorial/index.html is a useful place to start.

There is an interesting robotics research environment written in Python, called PyRobot, see http://pyrorobotics.com/?page=Pyro. It is an open source environment, and so, you can learn much from the (complex) code.

4.3 Computer Vision

One of the goals of the course is to introduce computer vision as a robot sensor. Cameras for use on robots are inexpensive and easy to setup. The challenge is to make use of the images and video that they provide. The data volume is large and onboard computers of the type we use are not very powerful.

Open Computer Vision (OpenCV) has been ported to Python and to the Raspberry Pi. This is a useful toolset for image processing. See the website:

https://opencv-python-tutroals.readthedocs.org/en/latest/#

Downloading and installing OpenCV on the NVIDIA TX2 computer is relatively straightforward.

This semester we will use a fisheye lens camera for image acquisition. It interfaces with the main robot computer via USB.

Figure 8 Fisheye lens camera (http://tinyurl.com/harly7g)

5 Robot Dynamics and Control

5.1 Basic Motion Control

Since it has two independently driven wheels (Figure 1), the robot is easy to maneuver using *differential steering*. (However, it is tricky to get it to go straight since the motors are not perfectly matched.)

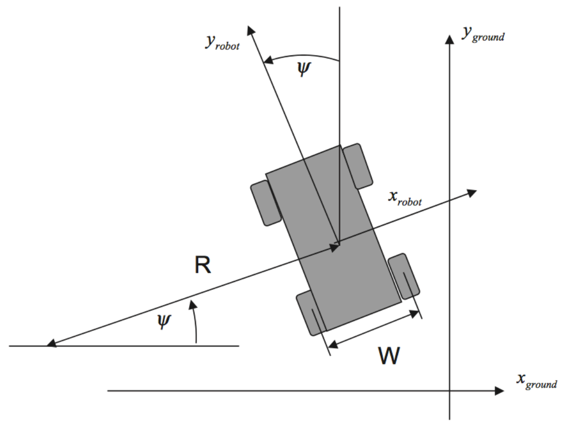

Figure 9 Coordinate system for a differential drive robot [2]

Referring to Figure 9, in *xy*-coordinates the equations of motion are

$$\dot{x} = -\frac{v_R + v_L}{2}\sin\psi$$

$$\dot{y} = \frac{v_R + v_L}{2}\cos\psi$$

$$\dot{\psi} = \frac{v_R - v_L}{W}$$

In this model the wheel velocities are the controls. Since it may not be possible to change the wheel velocities instantaneously, there can be additional equations for the velocities

$$\dot{v}_R = -\alpha v_R + u_R, \dot{v}_L = -\alpha v_L + u_L$$

Here α is a constant representing the inertial mass of the wheel and u_R, u_L are control torques applied to the individual wheels. Maneuver control involves selecting the controls to cause the robot to follow a path. Using the sensor readings, one can control the robot to avoid obstacles.

The paper by Thrun [4] is a useful survey of mapping methods using autonomous robots.

5.2 Behavior Based Robotics

One of the themes/methodologies we would like to explore this semester is use of "behavior" models for robot control and interaction. Some useful references are [1]–[3]. In a generally sense, the robot is trying to "learn" the behavior of its subject, and it should tailor its "behavior" to match the learned needs of the patient.

It is also important for the robot's subject to "trust" the robot. Trust is an important element in robot – human interaction. There are many papers on this, and we encourage you to think about this.

6 Deep Learning for Robot-Person Interaction

The JETSON TX2 is ideally suited for machine learning applications. In our project the robot should "learn" about its subject. Each team should investigate some aspect of machine learning in the project. For example, recognizing a specific individual based on facial recognition, gait, or other feature, or combination of features. It would be very useful to determine a subject's posture, speed over ground, etc. when the subject is moving in an area. We will discuss this topic more as the course develops.

References

References will be provided on the class website.