Learning Objectives

1. Automate path following (line following) procedures using computer vision (pixyCam).
2. Implement decision making through use of a state machine.
3. Design and control a mechanism to engage targets.
4. Incorporate closed-loop control into mechatronic applications
5. Prepare technical reports that include analysis of empirical data and evaluation of design options.

1 Overview

Autonomous vehicles have a wide variety of applications and are becoming more common. The use of robotics in hazardous environments provides a way for tasks to be carried out in a controlled manner without the need for a human operator. Some prime examples of this concept:

Husqvarna Demolition Robots
General Atomics MQ-9A Reaper
NASA's Mars Curiosity Rover
Boston Dynamics Spot

These systems carry out tasks with little to no guidance from operators. Each example must be able to navigate its environment, identify scenarios within that environment, and take an appropriate action.

In this project, you will design and build a system that can navigate a path, select directions in the path based on external markers, identify multiple objectives within the environment, and execute appropriate actions based on the present objective. The primary objectives you will encounter will require you to design and build two mechanical systems: a device to engage targets from a distance and a mechanism to deposit supplies.

1.1 Option: Propose your own project

Groups may also propose to conduct a project that differs from the course project. We expect these projects to be as, or more, rigorous than the course project. Details on this option, and on the project proposal presentation, are in Appendix D.

2 Project Description

2.1 Requirements

Develop a robot that can perform line following using the PixyCam. The instructors will provide a course for your robot to navigate; the full course will not be revealed until the day of the demonstration. The course will have several branches, each marked with a bar-code which will indicate whether your robot must take the left or right branch. Along the way, your robot will encounter either hostile or friendly units (designated by a color-code). Hostile units must be engaged from a distance of at least one foot, and the hostile unit is considered 'destroyed' once the target is knocked over. This will involve designing a device to launch ping pong balls, rubber bands, foam darts, etc at the target while maintaining a minimum distance. When your robot encounters a friendly unit, you must render aid by approaching to within six inches, playing a tone, and then returning to the course. You will encounter no more than three stationary hostiles and three stationary friendlies throughout the course. After carrying out these tasks your robot must continue the line following procedure until it reaches the end state. The bar-codes, color-codes, and an example course are shown in Appendix A and Appendix B.

In your report you will evaluate the performance of your robot in three key areas: line following, design of the 'shooting' mechanism, and target recognition/engagement. Line following should be assessed according to the characteristics we use to describe second order response (rise time, overshoot, settling time, etc). Evaluations of your 'shooting' mechanism should address the accuracy, precision, and range of the device. You may also assess the 'shooting' mechanism in terms of second order response characteristics. Target recognition and engagement will examine the amount of time taken to successfully complete an objective. A simple way to evaluate this is to create a log file that tracks the events that occur throughout the excursion and logs the time and state for each actions that takes place. For example:

$t = 6.83$ seconds, hostile target identified

$t = 8.38$ seconds, projectile fired

$t = 10.43$ seconds, hostile removed

$t = 12.84$ seconds, resume line following

2.2 Specifications

1. Your AUGV may be any ground platform, not larger than 18 inches wide x 18 inches long. Maximum height should be reasonable. Example platforms: TraXXas Stampede, MMP-5, Boe-Bot, AEK, KIPR.

2. Your AUGV must use a microcontroller- Raspberry Pi and other single board computers are not authorized (However, we do have some more powerful microcontrollers that you may use such as the Arduino Mega or chipKIT series). You may use multiple cameras and/or multiple microcontrollers.

3. If you choose to use multiple microcontrollers, then the microcontrollers **must** communicate with each other.

4. Your AUGV must sit idle at the start line until a push button initiates movement.

5. Your AUGV must use a Pixy2 to perform line-following and recognize different bar-codes/color codes. In Project 5 you used the 'Color Connected Components' class for the Pixy2 to recognize objects. The Pixy2 also has a 'Line Following' class that will recognize paths (including splits and intersections). Example code for basic line following operation is included in the Pixy2 Arduino library (Arduino IDE → File → Examples → Pixy2). You can learn more about the 'Line Following' class here.

6. While performing the line following procedure, your robot must travel no farther than 6 inches on either side of the line (relative to the outside of each tire).

7. The lines will be marked with black painters tape, which is approximately 5/8 inches wide.

8. Throughout the course bar-codes will communicate directions for navigating the course and alert your robot to the presence of an objective. The Pixy2 must be calibrated to recognize

at least 6 bar-codes while in the line-following routine (codes shown in Appendix A). The following commands must be carried out depending on the code detected:

(a) Turn Left

(b) Turn Right

(c) Go Straight

(d) Objective Ahead

(e) Bonus Objective Ahead

(f) Stop

9. When your robot is given the signal that an objective is present, your code must toggle the from 'Line Following' mode to 'Color Connected Components' mode. This will allow you to identify which color code is present, how large is it (how close it is), and where it is in the camera frame. One color-code will be associated with a stationary enemy, one with a moving (bonus) enemy, and the third will be associated with a friendly.

   (a) If an enemy is encountered, you must knock over the color-code by hitting it with a projectile from your shooting mechanism.

   (b) If a friendly is encountered, you must, without following a line, drive up to the friendly unit and stop 6 inches from the objective ($\pm 25\%$). Once stopped, play a 'tone' to simulate establishing communications with the friendly unit. After the tone is complete, return to where you departed from following the line, reacquire the line, and continue to navigate the course. Failing to approach within 6 inches ($\pm 25\%$) or running over the friendly units will result in a penalty (point deduction).

   (c) After taking the appropriate action and verifying the objective has been addressed, your robot must then toggle back to 'Line-Following' mode and continue to navigate the course.

10. Toward the end of the course will be a bonus 'Boss' enemy that is moving perpendicular to your direction of travel. Successfully engaging and destroying this enemy will result receiving extra credit points (amount TBD) on the final project. Once you have engaged (or bypassed) the 'Boss', continue line following. Once the end of the course is reached, the AUGV must come to a complete stop.

11. Your shooting device must be able to knock down the enemy target using ping pong balls, rubber bands, foam darts, etc as projectiles. An example of the target can be seen in Appendix A. You may not use explosives. You may not use metal projectiles. You may not use air soft or similar projectiles. You should limit the velocity of your projectiles to a safe value. Your device must be able to engage the target while maintaining a distance of at least 12 inches.

12. The dimension of the friendly and enemy objectives are:

   (a) Friendly and stationary enemies: two approximately 4x4 inch colored squares, surrounded by white-space such that the entire target is approximately 10" x 8"
   (b) Moving enemy: four approximately 2x2 inch colored squares, surrounded by white-space such that the entire target is approximately 10" x 8"
   (c) The center of the objective targets will be approximately inches above ground level
   (d) All objective panels are located in TH1136

13. Upload a video to Canvas that shows your AUGV satisfying each of the specifications listed above.

14. Prepare a technical report that, at a minimum, includes the following sections similar to a scientific publication: **Abstract, Introduction, Theory, Shooting Mechanism Design, Line Following, System Integration & Testing, Results & Discussion, and Conclusions & Future Work**. Note, this is a slight deviation from the report formats in Projects 4 and 5. You may also add additional sections and subsections as needed.

3 Notes and Hints

1. Restore your Pixy2 to a default known state prior to starting the Final Project. To do so, plug your Pixy2 into your computer via USB, then PixyMon → File → Restore default parameters.

2. We've found that the Pixy2 has more reliable line tracking and bar-code recognition when the camera is parallel to the ground. However, objective detection (color code detection) is more reliable when the camera is perpendicular to the ground. Therefore, we recommend either:

   (a) Mounting the Pixy2 on a pan-tilt mechanism
   (b) Using two separate Pixy2 cameras

3. It is possible to use multiple Pixy2 cameras with a single Arduino. You can either use multiple communication protocols (such as SPI and I2C), or you can use a single synchronous communication protocol (SPI or I2C) with multiple peripheral devices. However, the default Pixy2 Arduino cable and connection technique has a few issues:

   (a) Only pins 1-6 are pinned out at the Arduino end of the connector. This means that the SPI SS pin and I2C SDA pins are not available on the Arduino end of the connector
   (b) The Arduino ICSP header does not include an SPI SS pin

4. Details on using communication protocols other than SPI without SS (the default) with the Pixy2 are located here. There is also example code, `ccc_i2c_uart.ino`, located in the Pixy2 Arduino library (Arduino IDE → File → Examples → Pixy2).

5. If you choose to use a single Pixy2, the Pixy2 cannot track lines and color connected components at the same time. In addition, you shouldn't toggle between these two modes at a high frequency (it would be best to toggle once a bar-code indicates that you need to switch modes). Example code for toggling modes based on a timer is located on the course website Resources page.

6. In Project 5 we use the 'color connected components' mode to output the largest block of a certain color. For the final project, we'll still use 'color connected components' mode, however the objectives are identified with color codes. This helps reduce the probability of false positives. You can read more on color codes here.

7. There are CAD models for the Pixy2 mount and the TraXXas plate on the assignment page on Canvas. You may modify the mounts as you want. Submit models to the instructors for printing.

8. Finally, spend some time tuning your Pixy2 for both object (color) detection and line tracking. You should take a look at the line tracking tuning settings located here.

4 Due Dates and In-Progress Reviews (IPRs)

4.1 Due Dates

The points available on each portion of this lab assignment are shown below.

Portion	Points	Due Date
IPR I	20 points	3 NOV
IPR II	35 points	9 NOV
IPR III	20 points	18 NOV
Project Demo	60 points	7 DEC
Final Report	60 points	10 DEC
Peer Review	5 points	10 DEC

4.2 In-Progress Reviews

Project scoring is based on the quality of the in-progress reviews (IPRs), the demonstration, and the final report.

1. **IPR I (20 pts) due 2359 on 3 NOV:**
   (a) Develop a state transition diagram that describes the behavior of your AUGV. This initial design should capture all of the behaviors that you expect your robot to execute. With this large of a problem, you may need to create multiple diagrams that describe subsets of the overall AUGV. Submit your diagram(s) on Canvas. You should continue to update this diagram throughout the course of the project and include the final version in your report.

   (b) Provide a Request for Information (RFI) list to the instructors to address any questions or to request clarification about the design requirements / specifications. Submit on Canvas.

2. **IPR II (35 pts) due in-class during LSN 25 (9 NOV):** (**Group** - 17 pts) Brief the instructors on your design for the shooting mechanism. If you choose to manufacture any parts, provide your instructors the relevant CAD files for the part or parts that you intend to manufacture. Brief your presentation in class and submit the presentation on Canvas. Optional: Demo a prototype of your mechanism. (**Individual** - 18 pts) Develop pseudocode or Arduino C code to implement the Finite State Machine that you designed in IPR I. Include any updates that you've made since IPR I. Your pseudocode or code should clearly show, at a minimum, your states, event checking routines, state transitions, and state actions. **Brief one of the two implementations** to the instructors, and both group members submit **individual** solutions on Canvas. **Constraint:** Your total briefing time is a maximum of 8 minutes. **Note**: There is an example template for a state machine implementation in Appendix B.3 of your textbook.

3. **IPR III (20 pts) due in-class during LSN 27 (18 Nov):** Demonstrate an initial implementation of line following behavior and bar-code detection. At a minimum, your vehicle must be able to navigate a straight line, differentiate between right/left bar-codes, complete a 60 degree turn, navigate an additional straight line, recognize a stop bar-code, and then come to a stop. Verbally discuss your current closed-loop control implementation with the instructors. Your discussion should include the type(s) of controller(s), gains, fuzzy input/output sets (if appropriate), and tuning.

4. **Project Demo LSN 29&30 (60 pts):** Demonstrate proper operation of your project to the instructors and class during your assigned class hour. The demonstration on LSN 29 is optional.

5. **Final Report Submission (60 pts) due NLT 2359 10 DEC:** Submit the final project report and video to Canvas.

6. **Peer Review (5 pts) due NLT 2359 10 DEC:** Complete the assigned peer review, assessing your and your partner's contributions to the project. The five points are awarded for completing the peer review. Peer review feedback may be taken into account by the instructors when assigning final project and final course grades.

A Bar-Codes and Color-Codes

The bar-codes used throughout the course are available as individual PNG files and the color-codes used to identify friendly and enemy units are given as PDF files (both posted on the Final Project page here). The 6 bar-codes shown in Figure 1 correspond to the commands referenced in the Specifications section. Elongated versions of the barcodes shown are also available on Canvas.

0: Stop 1: Turn Left 2: Turn Right

3: Go Straight 4: Objective 5: Bonus

Figure 1: Commands and Prompts.

The color-codes used to define friendly and enemy units are shown in Figure 2.

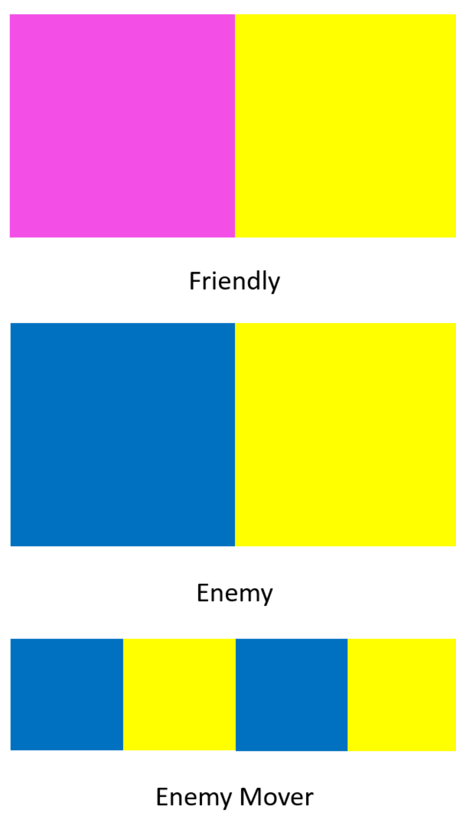

Figure 2: Identifications Markers.

Each color-code is 8” long (stationary 4” tall, moving 2” tall). All targets are printed on an 8.5” by 11” sheet of paper (you do not have to hit the color-code to knock over the target).

B Example Course

The course will follow the order of operations outlined in the Specifications section. The final layout will not be shown until the day of the demo, but you can expect any combination of the features shown in Figure 3.

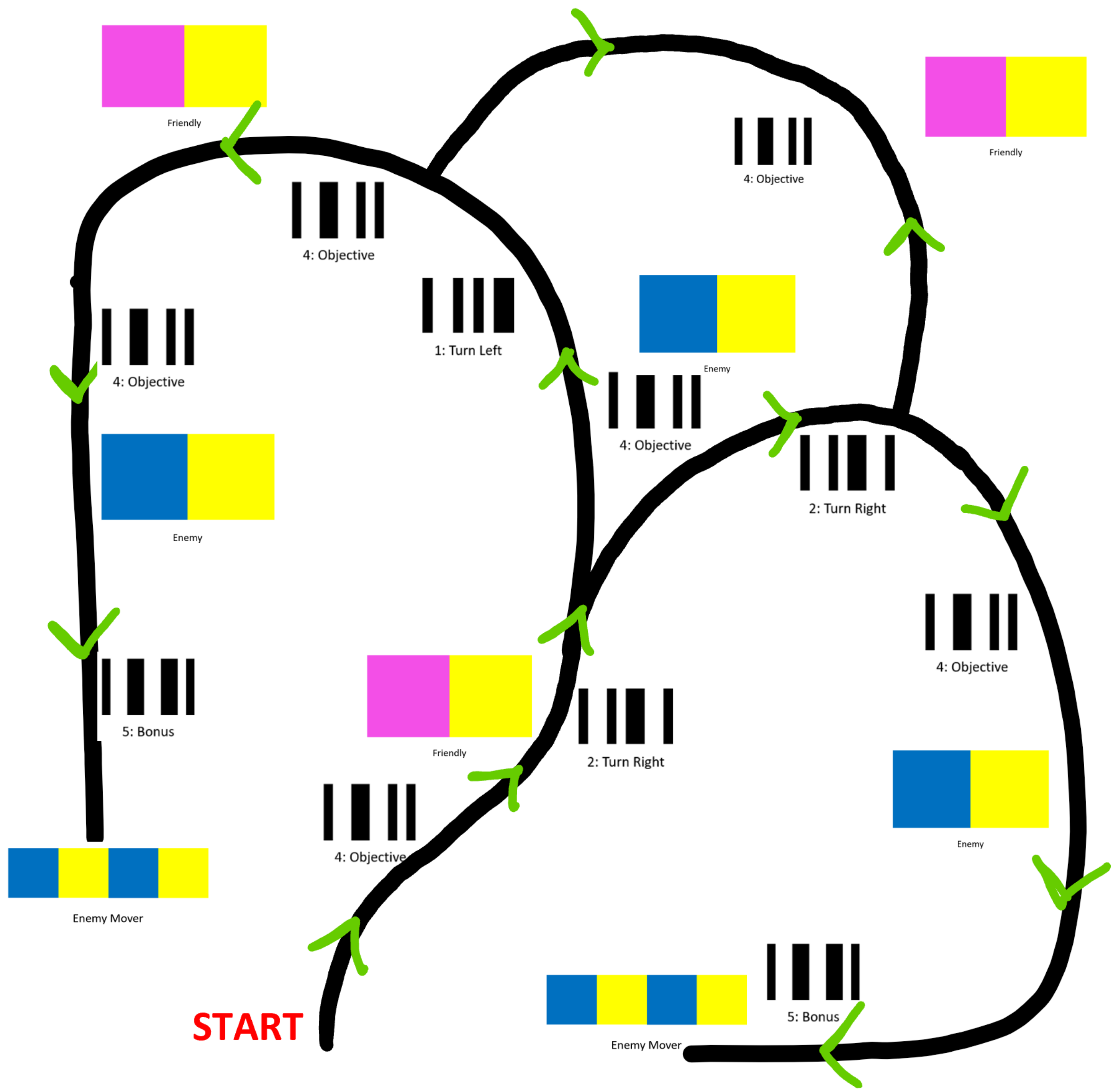

Figure 3: Example course.

C Tips for XE475 final report and common errors to avoid

- Write for your target audience: in this case you should write for undergraduate engineering seniors who have not taken controls or mechatronics. Therefore explaining the mechanics of PID control, Fuzzy control, sensing, and actuation is appropriate.[1]
- Be precise in your language. It takes many revisions to produce a good report.
- Write in the present tense where appropriate: "The system uses PID control."
- Write in past tense where appropriate: The control gains were tuned using a modified Ziegler-Nichols method."
- Write in future tense where appropriate: "This project will be repeated next year."
- Avoid step-by-step instructing like a lab manual.
  - Poor: "First write code that initializes the system with a pushbutton. Next write code that reads the magnetometer and calculates the error…"
  - Better: "In order to test and tune the Fuzzy controller's throttle response to a step change in distance, we placed a stationary leader vehicle five feet in front our Traxxas, initiated the test, and collected data as our Traxxas settled to the setpoint of three feet."
- DO include annotated images, schematics, and/or block diagrams of the physical system.
- DO include discussion of how you overcame challenges (e.g. multiple course branches, etc.), and include small segments of code in the body of the report to illustrate how you overcame these challenges.
- DO include equations where appropriate, e.g. control effort for PID control, transfer function for sensors, etc.

C.1 Plots

- Units are required. Title is required. Legend is required for multiple curves. Caption is required.
- The data or curve should boldly stand out—the grid should be subdued.
- The size of the figure can be rather small. The example in Figure 4 is 2.5 inches wide. It is still easy to read even at this size.

An internet search using the phrase 'producing publishable quality figures from Matlab' returns a number of high-quality websites/papers/presentations that deal with this issue. Some of my favorites are:

- Creating high-quality graphics in MATLAB for papers and presentations
- Preparing Figures in Matlab and LATEX for Quality Publications
- Publication quality figures using matlab: Why I do as much as possible within matlab
- export_fig.m

[1] Thanks to Dr. John Rogers for drafting this section, Dec 2017

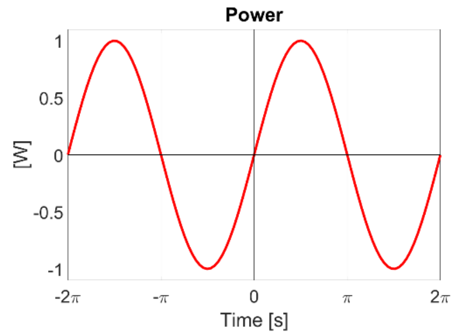

Figure 4: Example of a plot. In this case the sine curve is plotted in red, and the line weight is thicker than the grid. Units are included

D Optional Project Proposal

The course project is listed as one possible final project for XE475 and serves as the template. However, in the spirit of innovation and academic freedom (plus, just plain fun) we are open to groups proposing their own project. General guidelines, details on the proposal presentation, and some suggestions follow.

D.1 General Guidelines

In general, we expect the proposed project to require as much effort as the course project. It should be some type of Mechatronic system that incorporates a microcontroller as the primary computational device. It does not have to be a wheeled mobile robot.

- Must include some aspect of digital control (PID, Fuzzy, State-Space, System Identification, etc)
- Must present IPRs, similar to the course project
- Must use a microcontroller- Raspberry Pi and other single board computers are not authorized (However, we do have some more powerful microcontrollers that you may use such as the Arduino Mega or chipKIT series)
- Must incorporate one or more sensors (the quadrature encoder counts as a sensor)
- Does not have to incorporate computer vision (however, you may use CV if you choose)
- Does not have to use the Stampede platform.
- **NO** quadcopters or UAVs. If this area interests you, do an EE/ME489 or find an appropriate capstone project.
- Similar to the course project, your report should include appropriate figures and data analysis.

D.2 Example Projects

- **Arduino Engineering Kit (AEK)** There are three total robots that you can build using the AEK. Propose a project that incorporates and extends on the AEK. In other words, simply following the AEK instructions for one of the other robots would not be considered to 'require as much effort as the course project'.

- **Model the Stampede** Using knowledge from dynamics, controls, or other courses, develop either an analytical or SimScape model of the Stampede. Design a controller using the model and compare the expected (simulation) output to the actual output.
- **Robotic Arm** There are multiple robotic arms in the XE475 locker in various states of disrepair. Propose a project that incorporates one of these robotic arms. Note, this project will likely require some self-stufy in inverse/forward kinematics.
- **Propose your own** These are simply some example projects. Be creative and propose your own unique project.

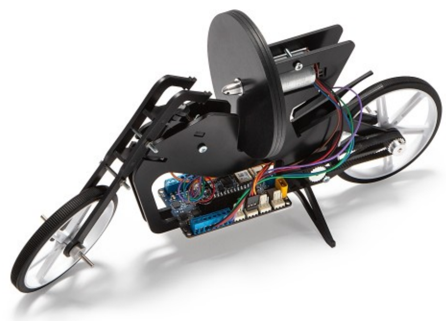

Figure 5: AEK motorcycle.

D.3 Project Proposal Presentation

Present a 10 minute (including questions) project proposal presentation to the course instructors NLT lesson 22. Coordinate with the course instructors if you wish to present earlier. You may want to briefly talk to one of us to give a general overview of your project concept prior to the proposal. The outcome of the presentation will be one of:

1. **Approved:** Proceed with the project as briefed
2. **Refinements required:** Proceed with the project once you have addressed feedback from the instructor(s)
3. **Denied:** Complete the standard course project

At a minimum, your presentation should include the following:[2]

1. Problem Description (the "Why?")
    - What are you going to do?
    - Why is it interesting?
    - Why is it hard?
2. Objectives (the "What?")
    - Define the scope of the work and clearly state the objectives, such as:

[2]Adapted from "Guidelines for Project Proposals", available from: http://writing.engr.psu.edu/workbooks/proposal_guidelines.pdf and "Computer Science 598d Course Project Proposals", available from: https://www.cs.princeton.edu/courses/archive/spring98/cs598d/projects/proposals.html.

- Design specifications
- Critical design issues
- Constraints, limitations, etc (we have imposed some of these on you already)

3. Approach (the "How?")
    - What approaches have others used?
    - What approach(es) are you going to try?
    - How will you analyze the performance of your solution?
    - What problems do you anticipate? Where may you need help?
4. Deliverables
    - The standard course project has a list of deliverables (IPRs, Final Report). How will your deliverables differ from the standard course project?
5. Summary and Questions