EE31 Junior Design
Spring 2018
Project Description

Assignment Date: 1-18-18 Due Date: 4-24-18

A working prototype is the expectation.

Background: ***New York Times, by Clive Thompson, November 8, 2017:*** **"**Street signage is the iconography of the automobile age. It's like highly functional pop art: silhouettes of schoolchildren, white arrows, rectangular cries of WRONG WAY and, most central of all, the ubiquitous stoplight. The traffic light might be the first part of that iconographic world to be transformed, or vanish altogether, once we are fully in the age of autonomous cars. Robots, after all, won't need signs to optimize the way they move through urban landscapes."[1]

"*An autonomous vehicle"*

Autonomous vehicles include cars, trucks, trains, buses, and other transportation venues where a human driver is not required. Also known as the self-driving car, the vehicle has the functionality of awareness of its surroundings in an arbitrary environment and navigating without human interactions from a starting location to an ending one. Alphabet (the parent company of Google) announced on November 7, 2017, that its Waymo division is actively testing "driverless cars without a safety driver as the driver position" [2].

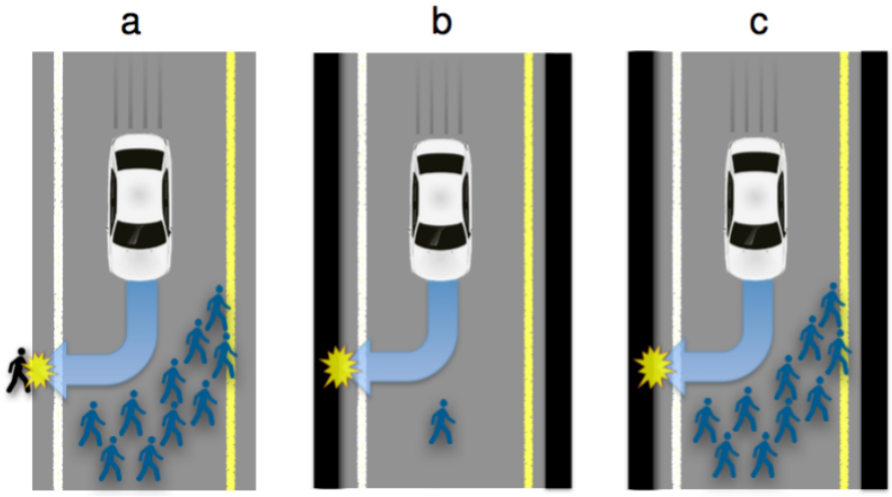
It's happening. Driver's licenses may be a thing of the past for humans. This is yet another disruption to society caused by the domination of technology in all our lives. The system architecture of an autonomous vehicle is rather simple: a collection of sensors to detect the vehicle's environment and identify obstacles; a computer system to interpret the sensor signals, define the navigation path, and make decisions of what to do; a control system for steering, speed, and monitoring system parameters; a drive system to provide motive force and stopping mechanisms; a communication system for data exchange with

other vehicles and network entities; user interface to display status to human occupants and attendants; and, a power system to distribute energy as required to each sub-system. The distinguishing feature and functionality of autonomous vehicles is the integrated control hardware and software systems—the intelligence—that replace the human operator. These systems evaluate the real-time data signals from sensors to keep the passengers inside the vehicle and any persons outside the vehicle safe from harm. Safety is the key to acceptance of this new technology and its prospective benefits to society. Speculation as to the benefits includes reduced costs for vehicles, transportation infrastructure, and healthcare due to minimizing accidents; further, this technology offers improvements to safety, satisfaction, and mobility for disabled persons. If worry-free transport happens, productivity increases are likely, as humans freed from driving responsibilities can work or play within the vehicle as they travel. Lastly, autonomous vehicles will spawn new business ventures and opportunities.

Street signage that will disappear from the landscape when autonomous cars are dominant, Illustration by Tamara Shopsin. https://www.nytimes.com/interactive/2017/11/08/magazine/tech-design-autonomous-future-cars-100-percent-augmented-reality-policing.html

However, the adoption of autonomous vehicles by society is far from assured. There are many significant obstacles to overcome: the technology challenges of reliable operability, safety, liability if some harm occurs, and possible resistance by people who like to drive. Autonomous vehicles cannot just replace all current vehicles on the road that require human drivers. Human drivers and autonomous vehicles will share the road for some time and this might just be somewhat problematic. Autonomous driving systems must satisfy a legal and insurance context to resolve a variety of issues both known and unknown. And lastly, there is computer security. With the rash of daily disclosures of hacking, a networked autonomous vehicle is a prime target. It is not clear how society will deal with computers that can roam among humans doing their own thing, or worse, doing something controlled by a hacker.

The barriers to acceptance for autonomous vehicles create numerous ethical issues, which are coming to the forefront as this technology emerges. While there is a strong belief that these types of intelligent vehicles will improve safety by reducing accidents—a current investigation suggests 90%[3]—that has yet to be determined as this number is not 100%. Some serious fatal accidents have occurred recently [6]. Yet, there are many potential benefits including providing a new level of accessibility for the young, old, and disabled. As such, what decisions must be made before autonomous vehicles take to the roads? And what decisions does the autonomous car make when it is on the road to avoid a possible accident? What are the legal, financial, criminal, and moral responsibilities for a fatal accident? Who is at fault? What are the privacy issues surrounding an accident and a fatal one? What responsibilities are necessary to provide for the injured person and their family? What if there is long recovery to full health, or job loss? Who will cover the impacts? There are many more questions to be posed and answered.



The Ethics of Autonomous Cars – "…they can never be perfectly safe. And that raises some difficult issues. How should the car be programmed to act in the event of an unavoidable accident? Should it minimize the loss of life, even if it means sacrificing the occupants, or should it protect the occupants at all costs? Should it choose between these extremes at random? The answers to these ethical questions are important because they could have a big impact on the way self-driving cars are accepted in society. Who would buy a car programmed to sacrifice the owner?" [17]

Source: https://www.technologyreview.com/s/542626/why-self-driving-cars-must-be-programmed-to-kill/

While engineers are good, they are not perfect. Poor designs, malfunctions, or construction flaws may cause an accident. Are the designers to be held responsible as manufacturers are so today? Yet, the history of car recalls, and the time to resolve such issues suggests, autonomous vehicle scenarios may be much more complicated. Several experts have suggested that autonomous vehicles must have an ultimate behavior scenario in case of potential disaster. This behavior must be programmed into the vehicles so humans know ahead of time how the vehicle will perform when a hazardous and accidental situation is detected. Here is the moral dilemma conceived as a thought experiment by Judith Jarvis Thomson in 1985 and published in the *Yale Law Journal* as

"The Trolley Problem":

"Suppose you are the driver of a trolley. The trolley rounds a bend, and there come into view ahead five track workmen, who have been repairing the track. The track goes through a bit of a valley at that point, and the sides are steep, so you must stop the trolley if you are to avoid running the five men down. You step on the brakes, but alas they don't work. Now you suddenly see a spur of track leading off to the right. You can turn the trolley onto it, and thus save the five men on the straight track ahead. Unfortunately, there is one track workman on that spur of the track. He can no more get off the track in time than the five can, so you will kill him if you turn the trolley onto him. Is it morally permissible for you to turn the trolley?"[4]

How would you answer?

As it concerns the designers of autonomous vehicles how does one program morality fit into a computer system or the hardware sensors to evaluate the environment accurately? What is the moral basis used by the designers to create the artificial intelligence? How is this moral basis translated to a computer system so it can make decisions? Whose moral basis is to be used?

The science fiction author Isaac Asimov presented *The Three Laws of Robotics* when he published "Runaround", a short story, in 1942 [5]. This story names three laws:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

Asimov's contribution to this ethical debate, or discussion, or investigation, is that society must agree upon the set of rules that define a set of behaviors for intelligent computer systems to follow in any and all situations. Privacy is one of them. Safety is another. Morality and ethical behavior are the most important. What policies need to be established to govern this emerging technology?

Assuming such moral and ethical issues are resolved, autonomous vehicles must collaborate. They are required to all work together as a constantly moving ad-hoc network exchanging information, command, control, and communications protocols to coordinate their movement and behavior. A transportation control center must also monitor and assess the network and its constituents to provide efficiency, performance, and safety. To do so, there are a lot of technologies that must be developed to have the autonomous vehicles interact, and additional technologies for the transportation infrastructure to support this new mode of mobility. Table 1 lists a sampling of these new technologies.

Sensors & microprocessors	Digital Signal Processing	Control Theory (analog, digital, adaptive cruise control)	Machine learning & Information theory	Short- and long-range radar
LiDAR – Light Detection and Ranging – for assembling a set of geo-spatial data points	Cameras and image processing	Ultrasound	Automated Highway Driving Assist (AHDA) (none, assistance, partial, conditional, high, full)	Vehicle connectivity & networking
Algorithms & robotics	Low-latency data transmission	Computer security	Fail-safe protocols	Manufacturing assembly & test
Collision detection	Logistic & mapping	User interface	Safety & accident protection	Battery & power efficiency

Table 1 – Sampling of Emerging Autonomous Vehicle Enabling Technologies

Autonomous vehicles have their evolution in robotics—the study of the design and manufacture of machines capable of performing a series of automated complex actions, typically under the control of programmable hardware and software as found in a computer. Swarm robotics is a new, rapidly growing field of technological engineering research to develop the hardware and software to coordinate multi-member distributed robot systems consisting of high numbers of (simply) designed (small and compact) robots. The objective of this research and design activity is to have collective behaviors emerge from the transactions among the robots and interactions of robots with the environment. Related to the engineering research is the concept of swarm intelligence, which builds upon artificial intelligence knowledge and evolutionary robots.
Autonomous vehicles that help one another navigate, provide safety, and share relevant information fall into this class of robotics.

A single member of a distributed robotic system is called a bot. When more than one bot works collectively with another member, the robots are called a swarmbot. When a large number of bots work collectively together to solve a problem presented to them via their interaction with their environment, the robotic system is called a community. This course uses these concepts and area of active research to simulate autonomous vehicles in an urban landscape.

Project Overview: The goal of this project is to model and simulate the navigation and environmental sensing of an autonomous vehicle by designing an intelligent swarmbot team using a community of two bots. The two bots are to work together, collaborating to solve the problem of safe navigation. You are expected to model, conceptualize, plan, design, build, test, and demo a two member swarmbot community that will simulate the navigation of two autonomous vehicles for a 6 hour time travel period (+1%, –10% hours). Additionally, the swarmbot must be able to position itself, for example, to come to the aid of a bot in case of an accident, communicate an emergency, or share relevant information about a situation occurring in the environment. To test that your team's swarmbot community meets these requirements, an acceptance test simulation, via three

course challenges, will utilize a smaller area and smaller time limit to scale the above limits.

The acceptance test simulation will be performed on an EE31 class supplied test platform. The entire platform will be used as the test environment representing a section of an urban landscape. Each bot will have ninety seconds to complete the acceptance test. Part 1 of the acceptance test is defined as: 1) each bot follows a simulated cityscape path from a designated start point to a designated end point; 2) the swarmbot is able to communicate with the Transportation Communications Center (TCC) to issue and follow commands; 3) the swarmbot is expected to recover from a collision, prevent accidents and provide alerts when such situations occur; 4) signal to human beings using visual indicators to indicate completion of tasks or state status; 5) the bots are expected to communicate among themselves to take up a position and signal start, finish, and appropriate information as necessary; and 6) the acceptance test is to be completed by the swarmbot within 90 seconds. (See Challenge 1.) Part 2 requires the swarmbot to position itself, perform some activity and movement, and then position itself at a particular location. For test purposes, the swarmbot must take instruction from its companion bot, to perform several movements, then return to the spot from which it started. (See Challenge 2.) The swarmbot must show novel functionality to enhance the behavior of the bot, or swarmbot community (see Challenge 3). And lastly, ***for each phase of the swarmbot project, your team needs to demonstrate the technical functionality of your hardware and software design to verify a detailed specification***. A technical specification has a value, units (dimensions), and a tolerance, for example, 10.00 (± 0.25) Volts, or 12.0 (± 2.0) Amps, or 8.75 (± 0.50) feet. The technical specification comes from one of the customer requirements. For example, the customer requirement is to complete the challenge in 90 seconds. Your team estimates that the distance the bot will travel along its path during the challenge is between 120 and 144 inches, or on average (nominally) 132 inches. Therefore, the technical design specification is that the bot must have an average velocity of 1.46 (+0.14 / −0.13) inches/sec to successfully accomplish the challenge within the 90-second requirement. An acceptance test plan is a procedure for an experiment having a pass/fail criteria to validate the design. For example, your team decides that testing the bot by having it move in a straight line for 18 seconds (20% of the time limit) is a reasonable estimate to determine success and the bot must do so for 8 or greater times out of 10 trials. If you perform the test (experiment) ten times and the bot is within spec for 9 of the 10 trials, then your team has verified this phase of the design. (Actually, only 8 successes were required, but the ninth success gives a bit more confidence in the result than the bare minimum!) A system acceptance test plan contains more tests and is more exact in their procedures and pass/fail criteria. You will do a more involved system acceptance test plan in Senior Design, but this exercise will introduce the concept.

Collisions with external objects are accidents and are to be prevented. One of the greatest dangers associated with autonomous vehicles is a collision with a pedestrian. One of the challenges for this project is to develop the ability to track a path to reach a desired destination without incident with your two-member distributed swarmbot system working as a swarmbot community. In our model, the swarmbot detects a colored paper path

representing the GPS, cellular, or Wi-Fi network navigation tracking data, communicates its location when it reaches intermediate waypoints, performs start-up diagnostics for performance and collision detection, identifies pedestrians, signals to TCC according to the safety protocol, and completes its travel. The swarmbot has visual indicators for visual feedback to indicate its situation, state, and to alert the authorities in case of trouble. The authorities, in this case, are the course instructor and the graduate TA.

For success, the autonomous vehicle simulation requires a photo-reflective sensor sub-system able to differentiate reflective pieces of colored papers on a black ground plane located beneath the bot. The black color of the ground plane may vary, so this needs to be taken into account. This provides for traveling along a route in a cityscape. Second, it is important to model an accident or the possibility of an accident of an autonomous vehicle with a pedestrian. The simulation uses a magnetic field and plastic obstacle to do so. A Hall sensor is utilized to detect the presence of a pedestrian (*South Pole of a magnetic field*). Since a magnetic field has polarity, this must be taken into account. Third, an ad hoc network is imperative to transfer relevant and timely data about changes in the environment. As such, the two bots that form a swarmbot community must communicate with each other in real time and with the TCC during the acceptance tests at specified locations. High-frequency sound is used as the communications signal.

In the acceptance test, the paths are constructed of specifically selected blue, red and yellow papers. All are fixed to a black ground plane. The swarmbot will be judged on its ability to maintain a tight tracking sweep (from side to side) as it traverses the multi-colored path.

A test version of the TCC is available during development. If the test set or real TCC is damaged in any way, physically or electronically, this will impact your grade. You need to treat the TCC with care and respect as any piece of test equipment. Many hours of thought and work were required to design, build, and test it.

The swarmbots are to use an Arduino Mega 2560 microprocessor board as the microprocessor platform. The Arduino software version for this course is ARDUINO 1.8.5. Other versions of the software may not be able to meet the design requirements for the course. The design teams are to use a designated set of components to optimize the cost of the bots. Datasheets of the components are on the course trunk site. Any deviation from this set of components, or those readily available in the lab, require approval from the instructor. Additional components for the Go Beyond phase may be of the designer's choice. No outside open source libraries other than those supported or promoted at the Arduino website are to be used for the swarmbots. You are welcome to write your own library that you write yourself for the Arduino.

The bots must use serial communication via sub-fringe ultrasound using a carrier frequency of 18.75 kHz and multiple data rates for an AM on/off keying transmitter/receiver. The design teams are expected to have their swarmbot communicate via a protocol, and with dedicated message lengths of 200, 300, 400, 500 ms unit step functions. The 200, 300, 400 ms unit step functions represent message_1, message_2, and

message_3 respectively, and are reserved as specific sequences by the TCC. The 500 ms unit step function shall be for alternative initial or conclusive bot1 to bot2 communication, ignored by the TCC.

The bots ***are not*** to use the serial communication protocol resident on the Arduino for their bot to bot communication. The resident serial communication on the Arduino may be used as a design and debug tool.

More specifically, each design team will be responsible to build its own single bot and then have the bot work with the bot from another team to form autonomous vehicle swarmbot community team. The bot teams are expected to solve the stated problem with the swarmbot acting as one team.

Specific Project Challenges: Figure 1 shows the test track. The blue, red, and yellow curved lines denote blue, red and yellow colored papers representing a path in a cityscape. Note the ~~exact~~ blue/red/yellow paths in the figure below are representational and are not in the exact configuration that will be on the test track. The start positions for the two bots are shown as the white X's labeled 1 and 2. (Note: the X1 and X2 will not be on the actual track.). Rectangular cubical obstacles (having approximate dimensions of 3" x 3" x 6") that may block portions of the path are not shown; bots may collide with these rectangular obstacles and must navigate around them. A wall surrounds the gray rectangle perpendicular to the page. Design teams will have to take into account that the bots must recover from collisions with these obstacles. Note that the TCC is beyond the active test track area where the swarmbot must perform and communicate to the authorities.

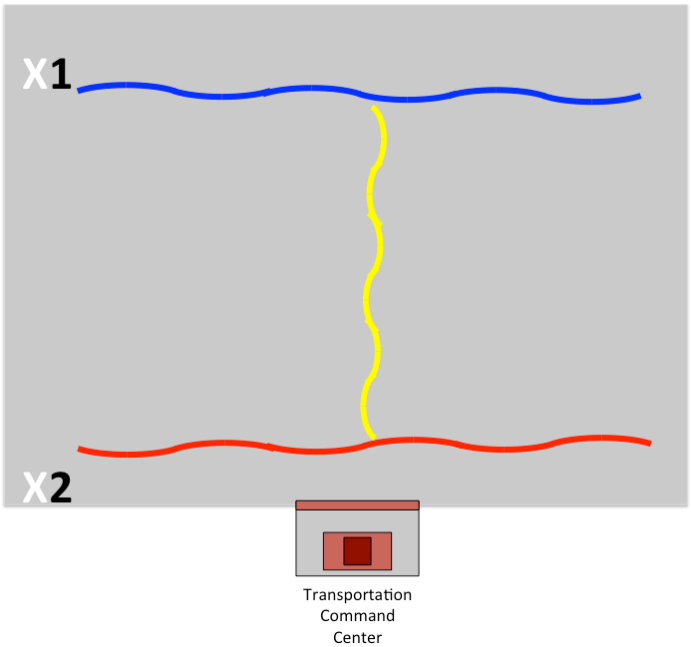


Figure 1 – Test Track

Figure 2a shows the tracking path for the first bot. The white dotted lines and other symbols are not on the test track. They are shown for clarification of the tracking problem that the swarmbot must solve. See Challenge 1 for more information about how the path is used for the course project. The Wi-Fi symbols (the multiple blue curved lines) designate locations where the bot must send and receive sound commands. The green pedestrian symbol designates the approximate location of the pedestrian. However, while the pedestrian will be placed as close to the path as possible, the figure indicates only an approximate location. The pedestrian symbol actually identifies two physical entities: 1) the stack of magnets generating a south polarity magnetic field perpendicular to the track and a plastic object near that location on the test track. The objective is to detect the magnetic field and not hit the plastic object. This action simulates detecting a pedestrian using an appropriate sensor and then avoiding a collision. Figure 2b shows the tracking path for the second bot.

Challenge 1: Bot 1 starts in the corner near the white "X1" in Figure 2a. Bot 1 flashes blue and red LEDs three times and signals to Bot 2 with a 500 ms message that it is starting. (Note: the messages sent by a bot is a unit step function that goes high at the start of the T ms, then goes low at the end of T ms, where T = 200, 300, 400, or 500. If the receiving bot evaluates the signal in 10 ms intervals, then the count at the end will be distinct enough to determine which message length was transmitted and received.) Bot 2 illuminates its communication LED, blinking it twice on receipt of the message. Bot 1 illuminates its start LED and then starts to move. The bot mimics the white dotted line until it collides with the wall of the test track. It rebounds from the wall, and then moves to find the blue path on the far side. When it does so, Bot 1 illuminates a blue LED. Bot 1 must follow the blue path until it detects the yellow path. Here it turns off the blue LED and illuminates a yellow LED. Bot 1 makes a 90° right turn on to the yellow path. The bot will be judged how tight a 90° right turn it makes. Bot 1 follows the yellow path until it detects the red path, when it turns off its yellow LED and illuminates a red one. Bot 1 makes a 90° left turn on to the red path. The bot will be judged how tight a 90° left turn it makes. Bot 1 travels along the red path until it detects the pedestrian. When it detects the pedestrian (magnetic field), it must flash a green LED, bring the bot to a full stop, and communicate with the TCC via a 200 ms message_1. When Bot 1 receives a responding 200 ms message_1 from the TCC, ***it continues along the red path to the end of the path at the wall. When it detects the wall at the end of the red path, it stops, blinking its red LED twice.*** Bot 1 rotates 90° to the left, stopping in the vicinity of the end red path and the spot on the wall where it collided, comes to a full stop and illuminates blue and red LEDs flashing them three times. Bot 1 then signals to the TCC with a 300 ms unit step function "message_2". The TCC upon reception relays this 300 ms message twice to Bot 2, with a 1 second delay, to enhance Bot 2's reception. This message to Bot 2 initiates its portion of the test, performing the same tasks just described for Bot 1 but on the red path and illuminating, flashing appropriately colored LEDs to indicate its status. Bot 2 signals Bot 1 that is it is starting to follow its path. Appropriate LEDs are flashed to indicate this exchange. When it discovers the pedestrian on the blue (magnetic field), it too communicates with the TCC with a 400 ms unit message_3, and illuminates its green LED in the same fashion as Bot 1. The TCC performs the appropriate emergency vehicle excitation, verbal commands and protocols, sends 400 ms

message_3 to Bot 2 to continue to the end of the path. ***When Bot 2 arrives near the end of the blue path, it too collides with the wall of the test track, rotates 90° to the right and comes to a complete stop, as did Bot 1, but blinking its blue LED twice.*** Bot 2 signals Bot 1 with a 500 ms unit step function "message_4" to start traversing an unmarked path in the black track, toward each other for a gentle collision. When they collide, the bots flash their headlights, braking lights, and turn signals ten times to indicate the end of the challenge. The LEDs must finish flashing to indicate the end of the test within the 90-second time limit. Additionally, there are several other behaviors the bots must demonstrate while traversing the path and challenge: 1) whenever the bots stop they must illuminate their red brake lights, until motion resumes when traveling along the path; 2) the bots must illuminate the appropriate yellow turn signals upon right or left turns; and 3) the white bot headlights must be illuminated for safety whenever moving or traversing the path.

After this first path traversal, the bots then switch their start positions and the test is repeated where the roles of the bots are switched. That is, each bot must be capable of switching roles with the other bot. Each test is ninety seconds.

Challenge 2: The two bots communicate with each other sending commands with a 500 ms message to have one of the bots move forward twelve inches, stop, turn around 180°, stop, move backwards three inches, stop, turn left, turn right, turn right, and turn right to end up as close as possible to the bot's starting position. A 500 ms message is sent to the other bot to perform the same test. This test is ninety seconds. Each bot has a designated mark on its exterior surface. At the start, the two bots have their marks lined up with each other. At the end of the challenge, the objective is to have the marks line up. The distance between the marks will be measured; the closer they are to each other the better the evaluation score. The objective is to make this distance as close to zero as possible.

Challenge 3: The swarmbot must drive at night (dim room lighting conditions) using only headlights (maximum two white LEDs) toward each other and avoid a collision. The test will be to place the bots at opposite ends of the longer dimension of the track. The two white diodes represented the headlights allowed as sources of illumination for this challenge. A photodiode, phototransistor, or photo resistor is to be used as the detector. When the two bots become close enough to detect each other's presence, they must stop, communicate, flash their headlights twice, and illuminate their rear yellow turn signals.

Challenge 4: The bots demonstrate their Go Beyond functionality. The bot teams may choose to do their own Go Beyond functionality or one that is shared among the swarmbot.

The last two classes of the semester are project challenge test days. Each team has only two attempts on each of the test days in April to demonstrate that their swarmbot has successfully met the three parts of the project challenge.

Go Beyond: Each design team is required to customize their bot or swarmbot with additional functionality to differentiate itself from other competitors attempting to obtain the purchase order from the customer. A Go Beyond proposal must specify the design, build plan, and test plan for the Go Beyond functionality of the bot. The Go Beyond proposal must be submitted by mid-semester to the customer (instructor) for approval prior to any work on the Go Beyond part of the project. The cost of the Go Beyond is considered as part of the overall cost.

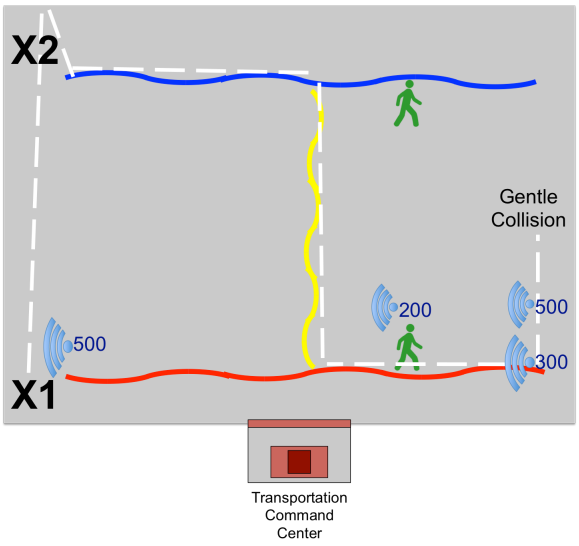


Figure 2a –Tracking Path for First Bot

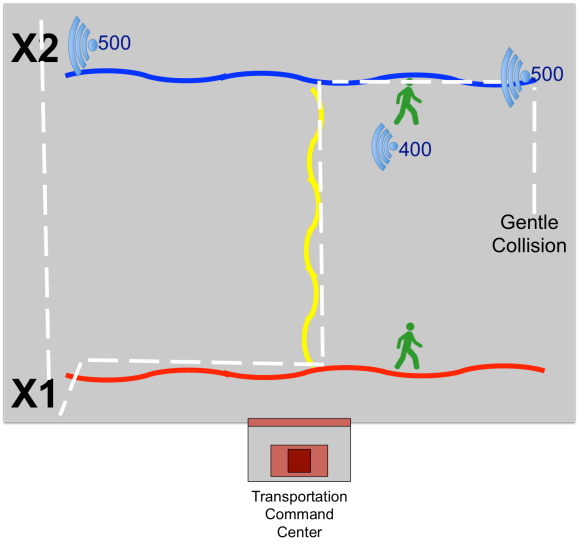


Figure 2b –Tracking Path for Second Bot

The purpose of these challenges represents a high technology company bidding on a very large and lucrative contract to supply manufacturers of autonomous vehicles with design services and sub-systems for their autonomous vehicles' customer. This exercise is to build a reliable demo to prove that the company has the capabilities to meet expectations. You will do the same to obtain the perspective of a design engineer in a company that develops products for commercial markets. Your team represents the company in the context of the course. As design engineers at the company, you are assigned to the project to build a prototype demonstrating the desired capability to potential manufacturers (customer) of the critical subsystems for an autonomous car. Since the manufacturing will be using the swarmbot throughout its development, you will need to estimate ongoing maintenance pricing should the swarmbot require field service repair. You will need to produce a sales invoice summarizing the swarmbot functionality, the cost of the swarmbot, and the cost for on-going customer service – Bot Care. The sales price of swarmbot must be based upon the cost of all the parts you used to construct your final swarmbot and the people hours it took to design, build, and test it. Finally, the final sales price cannot exceed a 65% margin on the parts and labor. An itemized parts list with parts cost and a time sheet of labor hours is required to justify your cost to the customer. The competitive cost for an engineer to build and test a swarmbot is $41.32 / hour. The competitive cost for a manufacturing worker to assemble the swarmbot is $26.23 / hour. Project manager competitive costs are approximately $65.15 / hour.

The Physical Design of the Autonomous Vehicle Swarmbot: Although this is a course in electrical and computer engineering, the physical design, appearance, and operation of the swarmbot is of significant importance equal to the hardware and software. There are no second-class citizens in Junior Design: the physical, hardware, and software are all first-class design challenges that you are expected to solve during the course of the semester. Note: all three design aspects are dependent upon each other's conception and implementation. Make note of the design trade-offs that you make in your notebooks, as this is an aspect that will be considered when evaluating your notebook and your swarmbot.

The Studio: Junior Design is a studio course with a team project. A studio course is characterized by having few lectures, students work in team to learn, activities and tasks emphasize and require collaborative and cooperative learning, instructors and TAs are on hand to mentor and guide through active questioning by students, active critique and discussion, reflection on failure to promote learning, responsibility for learning is on the student, responsibility for the instructors is to provide guidance, resources, and an environment that promotes respect and learning, and class tasks and activities are cumulative as the project progresses along during the semester. The strength of the studio is its dynamic learning and is integrated through immersion in content and context situations. This provides for personal intellectual growth.

Teams and Teamwork: Junior Design requires team collaboration. Teamwork is combined action and activity through the tasks and behaviors of the people on the team. The details concerning teams and teamwork have significant length. As such, it has been placed in the appendix. This is not because it is less important; rather this placement was

for clarity of this document to the reader. *Please read the appendix at the end of this document to understand your responsibility for the teamwork section of this course in the context of Junior Design.*

The Engineering Notebook: The instructor will assign each EE31 student to a team as a design engineer. Each design engineer must keep a record of his or her assignments and design drawings, analysis, schematics, parts, time, role, costs, and *reflections on experiments and failures* along the way, along with any other vital information, in a bound lab notebook. (One type of lab notebook is available from the bookstore, but there are others types as well. A bound notebook is one where the pages are stitched into the binding. This is ***not*** a spiral bound or loose-leaf style notebook). The notebook will be submitted at random times during the course, just before Spring Break, and at the end of the course. A grading rubric for the bot, the notebook, and reflections are available in another course document.

Project Executive and Executive Summary: Each EE31 student will be the project executive for two of the six design phases. Since you are a young company without a track record, your customer requires an executive summary to be submitted to the instructor and graduate TA at the end of each phase. The executive summary cannot exceed two written pages. A third page may be used for drawings, charts, schematics, etc., to convey visual design information. The executive summary must contain:

- A description of the deliverable, its functionality, and demonstration.
- The duration of the design tasks for the phase showing the original planned dates and any delays or changes from the original projections.
- A brief description of the conceptual and design activities.
- A short technology summary of the theory of operation for the team's design, any drawings, sketches, schematics, block diagrams, and flow charts to clarify the design that are appropriate to convey the design intent of the team
- A complete list of the experiments performed, the success or failure, an assessment of the quality of the reflection on failure, and the learning that occurred from the experiments. Consult the reflection on failure rubric to perform the assessment.
- A list of who was assigned which tasks and the quality of their work.
- The cost of the bot by phase to-date versus its estimated cost.
- Team stage assessment: strengths, weaknesses, difficult personality types present on the team, improvement plan, and success of respect and working together.
- Recommendations to the next project leader on how team operation and design work may be improved. For subsequent executive summaries after the first one, a statement of how the recommendations were implemented, or the reasons for discarding them must be included.

The executive summary must be submitted no later than one class after the team demonstrates the deliverable for the phase. Late executive summaries will not be accepted. The executive summary is not a team assignment. This is an individual assignment that counts toward the student's grade that is the author of the document.

You will be paired with another team building a second bot to form the swarmbot community. A third team will be assigned as an outsource partner to design and build a collision detection system for your team's bot. Your team is required to write a bot spec for the customer and a collision-detection spec for the outsource partner. Further, your team will need to have a cost estimate from the other team, and an invoice for the cost of the collision-detection system from your outsource partner. The cost of the outsourcing is part of your bot's cost. Part of your grade is to understand how to control design, development, parts, assembly, testing, and outsource costs. Otherwise, you will be unable to pass these costs on to the customer as part of the final price. Some portion of the customer's decision making on whether or not to purchase your swarmbot is based upon the cost of the bots. The instructor and TAs of the class represent the customer and are available to clarify and assist in the development of the working prototype. The swarmbot must be demonstrated and an invoice presented at the customer evaluation session during class on Tuesday, April 24, 2108. A second day of evaluation is scheduled for Thursday, April 26, 2018 to complete any demonstrations that may be required.

GitHub: All teams are required to have a GitHub repository (repo) for source code. We share a local server running GitHub at github.cs.tufts.edu. Use your EECS credentials to log in, so you need an EECS account (which you should all have). Invite the instructor and the graduate TA to be members of your team so we are able to read, review, and comment on the material placed in the repo. Every GitHub project can also have an associated Wiki, for other materials and an issue tracker for bugs and features. If you are unfamiliar with GitHub, you can download the entire **Pro Git** book, written by Scott Chacon and Ben Straub and published by Apress, available here: https://git-scm.com/book/en/v2. All content is licensed under the Creative Commons Attribution Non Commercial Share Alike 3.0 license.

Collision Detection Sub-system. Design Administration: Since collisions are bound to happen, each bot must have a collision detection sub-system. You are required to write a collision detection specification to define the performance, operation, and physical design of the sub-system.

Design involves transactions with other organizations to deliver complicated and innovative technologies. Your team will be assigned to contract with one of the other teams (not the team working on your companion bot) to design and build the collision detection sub-system of your team's bot. To do so, a methodology is presented to increase the chances for success and to reduce the risk associated with outsourcing this critical functionality:

1. Determine the objectives of the sub-system to be delivered: identify the critical requirements for delivery date, performance, how to accommodate changes in the design, sub-system level function, form, fit, acceptance test and debug, repair, and cost.
2. Define the inputs, the sub-system detailed functionality, and the outputs.
3. Identify the needs for each team, the constraints, and alterable items. Identify the operational requirements to which the outsourced sub-system must be interfaced.

4. Identify alternative methods/deliverables that will satisfy the stated objectives if obstacles and risks present themselves in the course of the design and development activities.
5. Write a specification for the performance and form/fit/function of your team's collision-detection sub-system. Have the outsource team sign and date the spec which indicates their understanding and acceptance of your team's spec.
6. Obtain a cost estimate from the outsource team and a final invoice of the cost of the collision-detection sub-system.
7. If companies do not perform, grievances can be worked out via the legal system. If any team is dissatisfied with the performance of the outsource partner, the legal process for redress is simulated by presenting the information to the instructor who will make a decision on the situation.

The instructor will assign the collision detection team that your team will work with to build your bot's collision detection subsystem, and which team's collision detection subsystem for which your team will be responsible.

A signed and approved collision-detection spec must be submitted to the instructor before work begins on this phase of the project. Both teams must jointly sign the spec. The spec must define a statement of work, an assignment of work, and cost estimates. If the actual development cost exceeds the cost estimate, this impacts the collision detection design team as additional cost to their own bot (you must recover this cost somehow and who else but the customer will pay for your mess up) and could impact your grade. A statement of work incorporates items 1 through 7 listed above, while an assignment of work defines a set of tasks, which engineer is responsible for the completion of each task, and the completion date. Spot checks to have the work explained are to be expected.

There are three additional assignments associated with the course: a pre-studio assignment to prepare your design considerations prior to building a subsystem, weekly status reports, and a project video.

Pre-studio Assignment: This is a set of questions, problems, or calculations for thought or solution to prepare your work in the design studio. The purpose of the design studio is to provide an environment where engineers can work together collaboratively, critique each other's work to improve upon the design, to understand what problem it is that is being solved, to learn the design method, and to validate the design, for if it doesn't work, it doesn't matter. The pre-studio helps toward the success of all these endeavors.

The Band Pass Filter Assignment: As part of the communications sub-system you are required to perform a MATLAB analysis for the second order band pass filter you design before constructing the sub-system hardware. Your team is to turn in your theoretical band pass filter analysis, MATLAB computations and m-file code, transfer function, frequency spectrum figures, and electrical schematics. The team must submit all the individual analysis of each team member and a short synopsis explaining the reasons which approach was selected. Each team member must record their approach in their lab notebooks.

Weekly Project Status Assignment: Each week the team is required to summarize what the team accomplished toward their project. This is to be in the format of a PowerPoint slide. This is limited to one PowerPoint slide. The format is: What tasks need to be completed this week? What tasks need to be completed next week? What are the risks? How are we behind? What do we need to do to catch up?

Project Management Assessment: Each team is required to mark the class poster with their completion date estimates and revisions. The instructor or TAs will mark the class poster with the date after the functionality demo is performed for that phase. Adherence to team estimated and class assigned completion dates is a factor in the class grading.

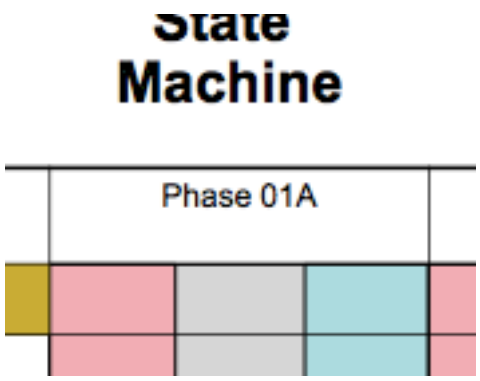


Figure 3 – Poster Project Management

Each phase of the project has an assigned start and due date set by the instructor. Your team is required to estimate your expected completion (due) date for each phase of the project. This is to be done on the class poster in the light red box. The date is designated as mm/dd. If you miss the estimated completion date, you have one chance to revise the estimated completion date, by entering a date in the gray box. The instructor or TA will fill in the date your team actually demonstrated the deliverable. Differences in estimates and completion dates are not without penalty. The penalty is $2000 added to the cost of your project. This ultimately will impact your grade.

Project Video: Your team will be required to submit a 90 second (+/-10%) video of your junior design project, posted to your project site for the course on trunk. The audience for your video is a prospective engineering student (17 or 18 years old) who is thinking about coming to the School of Engineering at Tufts. This was you three years ago!

The video must address and present the following information within the 90-second time length:

- The customer challenge and relevant context
- The problem definition of your junior design project
- The impact or value of solving this problem to society

- The coursework and engineering subject matter most relevant to you which enabled your team to solve the problem
- Any new competencies your team gained as part of developing the solution
- A demonstration of your swarmbot
- How the project helped your team to understand what an electrical or computer engineer does
- The video is to be submitted no later than Friday April 27, 2018 at 5 pm. The video is required to be submitted in an mp4 file format
- The videos will be graded on how well you address each of the bullets, captivate the interest of the intended audience, the genuineness of the video, and the quality of the video presentation.

Best is to start early. Contact Marc Raila, Digital Media Technologist at Tisch Library. He can be reached in the Digital Document Studio in Tisch, or via email at Marc.Raila@tufts.edu or by phone at (617) 627-0659.

Disclaimer: Clarification to the project and swarmbot specifications may occur during the semester that requires teams to redesign or modify the functionality of the swarmbot. Customer specifications change all the time in industry and government product development. A design engineer must be prepared for, and mitigate or implement contingency plans for such risks as they occur. This may impact the cost and the estimated delivery dates. The inability to adjust to such risks may impact your grade.

Delivery: All bots must be delivered to Miriam Santi in the ECE office no later than Tuesday, May 1, 2018. The bots will be stored in the display case outside the ECE office during the 2018-19 academic year. The bots will be returned to the teams at the senior dinner in May 2019. Upon return, the teams will decide on the disposition of the bots.

Lectures and Design Studio Sessions: The purpose of the lectures is to help shape the context of the swarmbot and its systems engineering. Your previous ECE courses provide you all the electrical and computer engineering material you need to successfully design and build the swarmbot. Explanations of technical concepts will only be presented as it relates to the swarmbot and its systems. If you do not have good ownership of previous course material, this is your chance to firm up those weaknesses while acquiring new design skills.

This course will begin to introduce you to teamwork and interpersonal skills. When you start your career as an engineer, or participate in an engineering summer internship, you do not choose the project or the teammates. EE31 is just like that. You are assigned to a team. Teammates do not always live up to expectations. They have lives outside of the project that interfere with their assigned tasks. Or, they may claim to be able to have proficiency in a particular area, but not to the level to actually successfully complete the task at hand. Team problems arise in all engineering teamwork. Leadership and

management require an honest assessment of the situation and confronting the adverse situation head on. You are not expected to possess such skills. The instructor and TAs have experience in these areas. Learn how to anticipate problems, or resolve them if they occur, by involving the instructor or TAs to help facilitate the necessary actions to reach a successful outcome.

This is a design course. Design is purposeful intent. It is about identifying a current problem state and envisioning a future desired state, then telling a story that describes each state, and the people, tasks, challenges, and path forward to transition from the problem state to the future desired state.

This is a design course. The expectation is that you will not understand much about the problem, the context, or the design methods at the beginning that are necessary for success. If you lack understanding or clarity, it is your responsibility to ask questions. You may not always get the answer you want. Your job is to discern what is important, what is not important, and make decisions on how to move down the road toward success. Your grade is based upon:

1. How well you define the problem
2. How well you interact with your teammates
3. How well you ask questions
4. Observations made by the instructor during your time in the studio sessions during class time or outside of class when the TAs will be in the lab for additional help
5. Your individual work product – your assignments and lab notebook
6. How well your swarmbot solves the problem

This is a design course. What problem are you trying to solve? If it doesn't work, it doesn't matter. It is about telling a story, where everyone in the story reaches salvation—they overcome their obstacles and challenges and reach what they deem success to look like and what they can take away from it. It is about purposeful intent. And when it doesn't work out, it is about how honestly you assess the situation and reflect upon the failure that occurs. It is about what you learn from your reflections. The ability to alter your perceptions to reality, and to get rid of the myths inside your head, are attributes of a good designer. Designers know it doesn't work the first time. And when it does, they know that they are missing something, because they know it doesn't work the first time. Design is purposeful intent. If you are designing you have a purpose, a reason, a motivation for your actions. If you don't, you are doing something else, and you are not doing design. This is a course on design. More importantly, it is about developing the courage to ask the questions that require asking.

Design Project Summary: The primary objectives of this project are to develop a bot and swarmbot in order to learn project design skills in preparation for the senior design project and integrate your knowledge of the fundamentals across the curriculum, while learning design techniques to construct a working prototype. As you progress through the course keep in mind your team will do best by giving attention to the ideas below.

- Understand the customer requirements – ask questions to determine any nuances or misunderstandings that might cause you to travel down the wrong design path.
- Understand the project requirements – ask lots of questions to be able to form a straightforward plan to accomplish the objectives.
- Research – realize that while you have been introduced to many engineering topics in the last two and a half years at Tufts, you may not have ownership of fundamental concepts. Ownership means that you have proficiency in the topic. For example, you are able to multiply large numbers 56789 x 12345 because you have ownership of the fundamentals of the multiplication tables 1x1, 3x4, 7x8, etc. If you need to review engineering topics from previous courses this is expected, as good fundamental knowledge will allow you to integrate engineering concepts across the curriculum. Further, if you are exposed to a new topic or concept, you are expected to learn this material on your own, or by asking questions to the instructor or TA. Design engineers ask lots of questions.
- Critique – students are hesitant to comment and critique another student's work in the presence of an instructor, as they believe any negative comments may impact the other student's grade. Engineers must critique the work of other engineers. The earlier in the design process the better, as changes will cost and impact the project less. This is called a design review. Critique in a class/design studio setting is expected for this course to improve design and learning.
- Physical design – the bot is required to have a clean, simple mechanical structure, well-dressed wires and cables, organized placement and assembly of parts.
- Physical size – the bot is to have mechanical dimensions to make it as small as possible, where boards and wires are packaged in an aesthetic appeal. Sloppy design does not cut it. The appearance of the bot has a strong impact on the team grade.
- Adaptable and flexible design – the bot design needs to be able to quickly incorporate additional sensors, processing, and functionality.
- Engineering – follow a methodical process; have a plan of action; and realize the technical concepts by implementing circuits, software, and mechanical parts, to accomplish a set of tasks.
- Energy – the bot needs to optimize battery life. An autonomous bot must be designed to have a long time between battery charges or replacement.
- Human factors – the bot needs to be small and easy to operate.
- Economical – the bot needs to have a high value proposition, i.e., the performance cost ratio is required to be a high number.
- Cost – the bot needs to be competitively priced to sell to a customer, otherwise a great solution will never make it into the marketplace.
- Documentation – document your work, keep track of your revisions, record all iterations, and be neat so others can read your writing.
- Teamwork – assign tasks, determine task durations and milestone completion dates, set expectations for quality of work and for the deliverable, make and follow-through on your commitments, and be realistic in what you need to learn, review, and accomplish.
- Collaboration – problems are more complicated than we initially perceive; the sophistication of technology does not allow an individual to have all the skills

required to solve the complex problems we face; we must collaborate—as it makes our tasks easier, provides for the opportunity to complete them faster; and, offers the opportunity to be better since there are more eyes for critique and review.

- Iteration – realize that nothing happens on the first attempt; set an objective and an outcome; if neither is realized then ask yourself what made you believe your course of action would work; learn and understand the cause of the failure; set a new objective and outcome on your path toward success.

Design Wisdom

- If it works the first time as expected, what did you miss?
- Model early, model often.
- Design with your imagination, build from a drawing. In other words, before you build, draw a schematic or make a plan. You will uncover your design mistakes earlier by doing so.
- Design errors and flaws are found faster with a schematic, than with a circuit of wires and components.
- Explain from a schematic (design language), demonstrate from a prototype (physical model). Use the two together to debug faster and more efficiently.
- If it doesn't work, walk around to the other side of the design bench and ask someone to explain to you what you did.
- If it doesn't work, what is not plugged in? If it doesn't work, why is it supposed to work? Why did you think it would work in the first place? What did you learn from it not working? What did you learn from making it work?
- Collaboration is the key to design excellence and innovation.

Team and Teamwork Appendix:

The importance of Team Work in Junior Design: Industry employers recognize teamwork as an essential skill for any new hire. The skills for proficient experiential learning, collaboration, and team cohesion are demonstrated through successful team projects. Teamwork proficiency requires understanding the technical and implementation challenges, integrated with the knowledge to transform complex problems into solutions making an impact on society.

Teamwork is not just about getting along with your teammates. Teamwork skills include planning and appraising, experimenting and making corrections, monitoring progress, navigating change, assessing risk, managing risk, conducting oneself ethically and professionally, articulating and expressing concepts and ideas—by verbal, visual, written, or demonstration venues, utilizing computer-aided design tools, and being respectful. Teamwork also requires certain actions and behaviors:

- Attending meetings on time
- Accomplishing work and meeting commitments on time
- Designing open-ended experiments, understanding the qualitative and quantitative data associated with the experimental outcomes, and reflection on the outcomes

- Performing at a high level to meet high standards of excellence
- Sharing ideas, credit, when things don't go right, working to get the project back on track when they don't, and in the success when they do
- Articulating and expressing ideas and concepts, or perspectives so all understand
- Reaching consensus—not my way or the highway—or at least an approach that you may not agree with, but are able to live with it
- Expressing your opinions and willing to listen to the opinions and suggestions of others
- Gaining learning and knowledge from each other
- Willing to help a teammate
- Asking for help from others on the team or beyond the team boundaries when necessary, when you don't know what to do
- Making commitments to the project goals and objectives
- Showing respect for others
- Knowing the difference of what is important and what is trivial—focusing on the important stuff
- Saying you don't know
- Asking questions when you do not know how to proceed or don't know the answer
- ***Offering no excuses***—being honest when you err, and taking corrective action for better performance in the future

There is a lot there and it is a big responsibility. However, all of this is required for success and for future continued employment!

Teams transition through five stages as teammates learn to work together effectively and efficiently. When you get a job in industry, you are assigned to a manager, a project, and a team. You typically have no choice, you get placed in the organization and are expected to ramp up your skills and familiarity to the project to reach the desired result. The development stages of teams are listed in the table below. While purposely picking the team is essential for success, this does not mean the team is ideal. It means that in someone's judgment, the people assigned to the team are capable of success. As you progress through this semester of Junior Design, note the stages of development your team passes through and how you have helped to enable each transition.

Stages of Team Development and Maturity

Forming – The team meets and learns about the opportunities and challenges, and then agrees on goals and begins to tackle the task

Storming – In this stage *"...participants form opinions about the character and integrity of the other participants and feel compelled to voice these opinions if they find someone shirking responsibility or attempting to dominate. Sometimes participants question the actions or decision of the leader as the expedition grows harder..."*

Norming – *"Resolved disagreements and personality clashes result in greater intimacy, and a spirit of co-operation emerges."*– This happens when the team is aware of

competition and they share a common goal. In this stage, all team members take the responsibility and have the ambition to work for the success of the team's goals.

Performing – *"With group norms and roles established, group members focus on achieving common goals, often reaching an unexpectedly high level of success."* By this time, they are motivated and knowledgeable. The team members are now competent, autonomous and able to handle the decision-making process without supervision. Dissent is expected and allowed as long as it is channeled through means acceptable to the team.

Outperforming & Adjourning – the team expands per the team's rules and what works for them to increase performance and success level. They harness their strengths and experiences as they disband the team. There is a time when all need to start on to the next project or move on to the next chapters of their own lives.

Source: Tuckman, Bruce W (1965). "Developmental sequence in small groups". Psychological Bulletin. ***63*** *(6): 384–399. doi:10.1037/h0022100. PMID 14314073.*

Along with development, leadership strategies change as well. What works at the beginning while the team is getting up and running and focused does not work when the team is in the thick of the implementation and time is getting short. The table that follows suggests some strategies that work as the team progresses along the development stages. As your team progresses through the semester make note of the strategies that work and don't work—reflect on why you had success or failure—and what you learned from the experience. How will this help you in your career?

Team Development Stage	**Leadership Strategies**	**Keys to success**
Forming (Setting the stage)	Coordinating Behaviors	- Purposefully picking the team; assigning roles and responsibilities - Facilitate team to identify goals - Ensure the team development of a shared mental model
Storming (Resolving conflict and tension)	Coaching Behaviors	- Act as a resource person to the team - Develop mutual trust - Calm the work environment
Norming & Performing (Successfully implementing and sustaining projects)	Empowering Behaviors	- Get feedback from the team and outside observers - Allow time for the transfer of leadership - Set aside time for planning and engaging the team
Outperforming & Adjourning (Expanding initiative and integrating	Supporting Behaviors	- Allow for flexibility in team roles

new members)		- Assist in the timing and selection of new member - Create future leadership opportunities

Source: *Manges, Kirstin; Scott-Cawiezell, Jill; Ward, Marcia M. (2017-01-01). "Maximizing Team Performance: The Critical Role of the Nurse Leader". Nursing Forum. **52** (1): 21–29. doi:10.1111/nuf.12161. ISSN 1744-6198*

Negotiation: Most times, even people do not realize it, their simple conversations are negotiations to obtain what it is they desire—at that moment, or in the future. Negotiation is an interaction between two or more people with an implied or explicit purpose to reach a favorable conclusion about something causing a conflict to exist—among the people or the issue at hand. The favorable result can impact one of the parties to the negation in a positive way, or to all, or to just a subset.

The purpose of negotiation is to resolve the differences concerning an entry or an exit, to gain advantage, or satisfy various constituents. The process typically begins by one party stating a position, then through discussion by making compromises and concessions to obtain a favorable path forward. Trust is the critical element in influencing an outcome and making the process of negotiation successful. Often divergent parties must cooperate to improve possibility for success. The bullet list below provides some guidelines on negotiation.

- Take an interest-based focus – concerns, desires, and needs of each team member
- People's preferences and priorities are learned via a formal meeting to exchange of information so that and trade-offs can be made to overcome barriers to create a win-win for each team member
- Define a positive team atmosphere to create the good dynamic for success.
- Define for each team member the balance between challenge and ability to achieve maximum performance for the team.
- What distractions could derail positive teamwork?
- How does each team member achieve satisfaction?
- Question each assumption.
- Ask for proof or clarification.
- Don't believe someone else's approach will work. Be skeptical and verify it.

Team Issues and Problems: People are the problem! Learning how to resolve people problems will make you a better engineer—as you are able to get things done with people. Authority and power are not the keys to success or resolving problems. The more effective you are at influencing those around you, the more successful you will be. However, you need to be aware of the problems that can arise within team and the personalities of team members.

Here are some issues that influence negative behavior and can cause problems:

- Competition – increases friction between team members – different interests, goals, or needs for recognition.

- Working behaviors differences – someone wrapped up in their own habits and behaviors will tend to oppose another team member's ways of working (my way or the highway).
- Ignoring normal team rules of the road – behavior inconsistent with expected actions may cause conflict.
- Inability to perform well or as expected (hare your part of the load) –when a team member is unable to contribute equally or whose performance or work product is below par, or unmotivated to work properly, other team members may become frustrated causing conflict.
- Lack of well-defined objectives and tasks – with a lack of focus fighting about what to accomplish or when team members have different ideas about how the goal should be reached—this causes conflict.
- Lack of accountability – lack of individual and team responsibility for task and objective completion results in stress and tension, leading to conflict

Difficult personalities: They are everywhere, so get proficient with how to handle them and deal with them, so they don't impact your desired outcome. The list comes from Brinkman and Kirschner [24] who define *The Four Intents* (which can lead to conflict in dealing with people). These are the fundamentals for understanding negative behavior:

- Get the task done.
- Get the task right.
- Get along with people.
- Get appreciation from people.

To deal with difficult people you need to understand their intent, that is, the root cause of the behavior. Identifying the root cause of the behavior can lead to change in the interactions with the difficult person. People have a pattern to their behavior. Observing them allows you to anticipate, interact, and react more effectively. Remember: you are not responsible for someone else's behavior; you are responsible for how you react to it. Note 1: the person's focus – task—getting something done; or people—getting along with others. Note 2: the person's activity – passive—their tendency is to be into the task details and to work alone; or be aggressive—by dominating the situation. The solution is to find a balance while keeping the negative behavior contained.

The 10 Behaviors: Brinkman and Kirschner describe ten difficult behaviors of normal people in a team environment and suggest techniques on how to overcome the obstacles to success that occur when these behaviors are present. The ten difficult behaviors are: *the tank person, the sniper, the know-it-all, the think-they-know-it-all, the grenade, the yes person, the maybe person, the nothing person, the no person, and the whiner*.

The Tank Person is confrontational, pointed and angry, the ultimate in pushy and aggressive behavior. Dealing with a Tank Person is to command respect.

Action Plan for Dealing with a Tank Person

- Hold your ground. Stay put and hold your position. Silently look the Tank in the eyes and shift your attention to your breathing. Breathe slowly and deeply. Intentional breathing helps you regain self-control.
- Interrupt the attack. The best way to interrupt anyone, whether yelling or not, is to evenly say their name over and over again, until you have their full attention.
- Quickly backtrack the main point. When you have the Tank's attention, backtrack (echo back) the main accusation. Be quick about it. The Tank is speaking and thinking at a rapid pace, so blend by speeding things up.
- Aim for the bottom line and fire! The bottom line varies according to your situation but it usually is about two sentences long. Preface your bottom line with ownership of it, by saying, "From my point of view …" or "The way I see it …" This prevents your shot at the bottom line from restarting the war.
- Peace with honor. Redirect to a peaceful solution by offering the Tank the last word, only you decide where and when.

The Sniper uses rude comments, biting sarcasm, or a well-timed roll of the eyes. Making you look foolish is the Sniper's specialty. Snipers take shots at you to make you look bad or to try to undermine you.

Action Plan for Dealing with a Sniper Person

- Stop, look, backtrack. Zero in on the Sniper. If it seems like someone is taking shots at you, stop – even in the middle of a sentence or word. Bring all your activity to a standstill. Scan for the Sniper and backtrack.
- Use searchlight questions. Use two questions to expose the Sniper's behavior: 1) Intent – "When you say that, what are you really trying to say?" and 2) Relevancy – "What does that have to do with this?"
- Use tank strategy if needed. Hold your ground, interrupt the interruption, backtrack the main accusation, and aim at your own bottom line.
- Go on a grievance patrol. If you suspect someone is holding a grudge against you, but you're not certain, go on patrol and see what you can scout out.
- Suggest a civil future. Finish the interaction by suggesting an alternative behavior for the future.

The Know-It-All is seldom in doubt, and has a low tolerance for correction and contradiction. If something goes wrong, however, the Know-It-All will speak with the same authority about who's to blame – you! The Know-It-All can be one of the toughest of all the types of difficult people to deal with. The challenge with a Know-It-All is that often enough they do, which perpetuates the pattern.

Action Plan for Dealing with a Know-It-All

- Be prepared and know your stuff. Clearly think through your ideas ahead of time. The Know-It-All defense system monitors incoming information for errors. Know-It-All will pick up any shortcoming and use it to discredit your whole idea.
- Backtracking respectfully. You have to do more backtracking (echo back) with a Know-It-All than any other difficult person. If you don't backtrack, you run the

risk of having to listen to the Know-It-All as they repeat themselves, over and over again.

- Blend with doubts and desires. If the Know-It-All has doubts about your idea, then there is specific criteria that aren't being addressed, such as the reasons why or why not. Show how your idea factors the Know-It-Alls specific criteria into account. Know-It-Alls tend to have a finite set of dismissal statements that reflect their highly valued criteria.
- Present your views indirectly. Proceed quickly but cautiously while defenses are temporarily down. Use softening words "maybe," "perhaps," "this may be a detour," "bear with me a moment," "I was just wondering," and "What do you suppose." Use plural nouns like "we" or "us" over "I" or "you." It can help give the Know-It-All a bit of ownership.
- Turn the Know-It-All into a mentor. Openly acknowledge the knowledgeable problem person as your mentor in some area of your life that you seek to develop. By letting the Know-It-All that you recognize an expert, and are willing to learn from one, you become less of a threat. You may find your way from the "disenfranchised" group into the "generally-recognized-as-safe-to-listen-to" group.

The Think-They-Know-It-Alls people can't fool all the people all the time. But they can fool some of the people enough of the time, and enough of the people all of the time – all for the sake of getting some attention. They know how to learn just enough about a subject to sound like they know what they are talking about. They are addicted to exaggeration as an attention-getting technique. They might even suffer from the Dunning-Kruger effect (dumb people don't know they're dumb.)

Action Plan for Dealing with a Think-They-Know-It-All Person

- Give the person a little attention. Use two ways: 1) backtrack their comments with enthusiasm; and 2) acknowledge positive intent rather than wasting your time with their content.
- Clarify for specifics. Ask them for some revealing clarification questions for specifics. Since the Think-They-Know-It-All speaks in huge generalizations you'll want to question the use of universal words like "Everybody" with "Who specifically?", "Always" with "When specifically?", and "Significant" with "Significant in what way, specifically?"
- Tell it like it is. Redirect the conversation back to reality.
- Give the person a break. Resist the temptation to embarrass them. Make them an ally by giving them a way out and again minimizing the chance of putting them on the defensive.
- Break the cycle. Recognize the negative cycle and work with the person to break the cycle. Break the cycle by doing two things: 1) use gentle confrontation to tell them the truth about the consequences of their negative behavior; and 2) actively look for and notice what this problem person is doing right, and give them credit where credit is due.

The Grenade Person after a brief period of calm, the Grenade person explodes into unfocused ranting and raving about things that have nothing to do with the present circumstances. Watch out. If you've ever asked, "Where did that come from?" you're not alone.

Action Plan for Dealing with Grenade People

- Get the person's attention. To get people's attention when they are losing control of themselves, call their names, raise the volume of your void so you can be heard through the explosion, and wave your hands slowly back and forth in front of you.
- Aim for the heart. Show your genuine concern for these problem people by saying what they need to hear. By listening closely, you can determine the cause of the explosion, then backtrack while reassuring them of your concern. When you hit the heart you'll be surprised how quickly the Grenade calms down.
- Reduce intensity. Reduce your volume and intensity.
- Time off for good behavior. Don't try and have a reasonable discussion about the cause of the explosion during the explosion. Take a time out, whether 10 minutes, an hour or a week, and then have a meaningful follow-up on the episode of temper.
- Grenade prevention. Find the pin and don't pull. If you can find what pulls out the pin on a Grenade, you can act to prevent it from being pulled again. If you have a good enough relationship, you could simply come right out and ask the Grenade what makes them mad. Start by stating your intent clearly, "I want to reduce the conflict with you." Then use clarifying questions to get specific on the cause of anger. A useful question is, "How did you know when to get angry?"

The Yes Person, in an effort to please people and avoid confrontation, says "yes." They say "yes" without thinking things through. They react to the latest demands on their time by forgetting prior commitments, and overcommit until they have no time for themselves. Then they become resentful.

Action Plan for Dealing with a Yes Person

- Make it safe to be honest. Make the communication environment a safe one so that the two of you can honestly examine whether promises being made in the future will be promises kept. This could be a one-time long conversation, or it may require several meetings over an extended period of time.
- Talk honestly. If you think the Yes Person is angry or resentful about something, or believes in the excuses, whether justified in your opinion or not, encourage the person to talk it out with you. Hear him out, without contradicting, jumping to conclusions, or taking offense.
- Help the person learn to plan. Once you've listened to your Yes Person's point of view, it will be obvious to you "why" you can't take "yes" as an answer. This is the time to create a learning opportunity. By using the past experience as a template, you can go back together and approach the task as if it's in the future. What motivation was missing? What could have been done differently? How else could the situation have been handled? Help the Yes Person focus in on the specific action and steps and process involved in accomplishing the task.

- Ensure commitments. At the end of the discussion, thank your Yes Person for talking the problems out with you, and ask, “What will you do differently the next time you’ve made a promise to me and you are unable to carry it out?” Once you’ve received your answer, you must follow through and ensure commitment. See *Five Ways* at the end of this section
- Strengthen the relationship. Look at every interaction as a chance to strengthen the relationship. Acknowledge the times when your Yes People are honest with you about doubts and concerns, make an event out of every completed commitment, and be very careful how you deal with broken promises.

The Maybe Person in a moment of decision procrastinates in the hope that a better choice will present itself. Sadly, with most decisions, there comes a point when is it too little, too late. And the decision makes itself.

Action Plan for Dealing with a Maybe Person

- Establish a comfort zone. Nebulous fears and negative feelings interfere with clear thinking. For example, if you ever told a salesperson you were going to “Think about it”, even though you knew you weren’t going "To buy it." The get along part of you didn’t want to deal with the discomfort of telling the truth.
- Surface conflicts, clarify options. Patiently explore, from the Maybe Person’s point of view, all of the options and the obstacles involved in making the decision, and any people that might be adversely affected by the decision. Listen for words of hesitation like “Probably,” “I think so,” “Pretty much,” “That could be true," and so on as signals to explore deeper.
- Use a decision-making system. The best way to make a decision is to use a system. if you have a system that works well for you, teach it to the Maybe Person. One approach is to list out all the positive and all the negatives explicitly
- Reassure, then ensure follow-through. Once the decision is made, reassure the Maybe
- Tell the person that there are no perfect decisions, and that the decision is a good one. Then, to ensure that the Maybe Person follows through, stay in touch until the decision is implemented. For example, “I’ll speak to later to follow up on this.” See *Make Decisions* at the end of this section.
- Strengthen the relationship. Promote the idea of a better future for both of you as a result of the person’s honesty with you. Be willing to take a few moments from time to time and listen to the Maybe Person’s concerns. Talk on a personal level with the person, and help him or her learn the decision-making process whenever the opportunity arises. If you are willing to patiently invest a little time in this kind of guidance, the Maybe Person will never want to let you down.

The Nothing Person doesn’t contribute to the conversation. No verbal feedback. No nonverbal feedback. Nothing. What else could you expect from … the Nothing Person? The Nothing Person can be especially dangerous if they appear as a Yes Person, but they actually are a “Say Yes, Do No” person.

Action Plan for Dealing with a Nothing Person

- Plan enough time. If you are tense and intense because of limited time, it's the wrong time. Plan ahead. Dealing successfully with the Nothing Person may take a long time.
- Ask open-ended questions expectantly. The best kind of question to ask a Nothing Person is one that can't be answered with a yes, a no, or a grunt. Use questions that begin with the words Who, What, When, Where, and How, since they tend to open up topics for discussion.
- Lighten it up. When nothing else is working, a little humor carefully used can go a long way. Be careful with this, because humor is a two-edged sword. It can inadvertently cut the Nothing Person and yourself, and there's nothing funny about that.
- Guess. If your Nothing Person has remained silent until now, and nothing else has gotten results, or you want an alternative, try this: Put yourself in the Nothing Person's shoes, and think back on the course of events as you understand them. What was the sequence and how else might you interpret the sequence to make positive sense out of this negative silence? Once you've come up with an idea, suggest it to the Nothing Person and watch for a reaction.
- Show the future. Sometimes, the only way to get a Nothing Person to talk is to take him out of the moment and into the future. There the Nothing Person can see the consequences of continued silence, and perhaps find enough perspective and motivation to open up.

The No Person kills momentum and creates friction for you: more deadly to morale than a speeding bullet, more powerful than hope, able to defeat big ideas with a single syllable. Disguised as a mild mannered normal person, the No Person fights a never-ending battle for futility, hopelessness, and despair.

Action Plan for Dealing with a No Person

- Go with the flow. The first action step for dealing with negative people is to allow them to be as negative as they want to be. The worst course you can take with negative people is to try to convince them that things are not so bad and could be worse. This only motivates negative people to work more intently to convince you that things actually are so bad, and will be worse. Put another way, attempting to convince a No Person to be positive is like struggling to climb out of quicksand: The harder you struggle, the more embedded you become.
- Use the person as a resource. The No Person can serve two valuable functions in your life: They can be your personal character builder, and they can serve as an early warning system.
- Leave the door open. No People tend to operate in a different time reality than other people. Any effort to rush them to a decision will force them to slow down. With enough pushing for action, No People will put enough drag on things to bring them to a complete stop, or become the sand in the gears that eventually destroys the motor. Whereas the temptation might be to throw them out, exclude them, or to think, and leave the door open so they can come back in when ready. For example, "If you change your mind, let us know," or "When you think of a

solution, get back to me," or "Why don't you think about this for a while, and report back any ideas you have."

- Go for the polarity response. There are two ways to apply the polarity principle when dealing with No People. The first is to bring up negatives before they do. The second is to just agree with the hopelessness of the situation, and take it one step further. This may cause the No Person to go in the opposite direction.
- Acknowledge the person's good intent. If you are willing to project good intent onto negative behavior, negative people may come to believe it. Then that analytic perfectionism can be expressed in a more useful way. Decide to act as if the negative feedback is meant to be helpful. Appreciate the No Person for having such high standards, for the willingness to speak up, and for the concern about details.

The Whiner --nobody likes a Whiner-- feels helpless and overwhelmed by an unfair world. They don't get that if you laugh the world laughs with you; whine and you whine alone. Their standard is perfection, and no one and nothing measures up to it. But misery loves company, so they bring their problems to you. Offering solutions makes you bad company, so their whining escalates.

Action Plan for Dealing with Whiners

- Listen for the main points. The last thing you want to do is to listen, but that's just what you need to do with the Whiner. Listen with pen and paper to catch the main points of the complaint.
- Interrupt and get specific. Take command of the conversation through a tactful interruption, and ask for your Whiner's help. Then ask clarification questions to get to the specifics of the problem, because vague problems are rarely solvable.
- Shift the focus to solutions. Because Whiners often complain in vague, cascading generalizations (e.g., "It's all wrong. But even if it wasn't nobody cares."), they don't stand still with any one problem long enough to stand a chance at problem solving. Once you begin to get specific about each complaint in turn, Whiners find themselves face-to-face with specific problems.
- Show the Whiner the future. When people have been feeling helpless, it is helpful to give them something to look forward to. If solving the problem they've brought to your attention turns out to be your responsibility, then you must keep your Whiners informed about progress.
- Draw the line. If backtracking, clarifying, and asking for a direction has not produced any real change in the Whiner, drawing the line becomes necessary. If your Whiner gets back on a roll with complaining, and it sounds like it isn't going to stop, take charge of the situation and bring it assertively to a close.

Five Ways **to Ensure Commitment and Follow-Through:** There are several things you can do to make someone more dependable. Brinkman and Kirschner suggest for task related projects:

1. Ask for a commitment. Ask the person to agree by making a commitment; a commitment deeper than just agreeing.

2. Ask your teammate to repeat the commitment back to you in their words**.** If a person repeats back to you what you said, it is provides both clarification and understanding. If the repeat doesn't sound correct, do it again until all are on the same page.
3. Write it down. Make sure they write down what you are asking to be completed before leaving. Writing down a commitment makes someone more likely to act on it.
4. Non-standard deadlines. Make a deadline like 10:19 am rather than 11 am. Making a deadline different and non-standard makes it stand out and is remembered.
5. Set negative consequences. As a team agree to the negative consequences of not following through on a commitment. Make it public beyond the scope of the team by informing the TA or instructor.

Making Decisions **that make an impact:** Gary Klein explains the Recognition-Primed Decision (RPD) Model in his book *Sources of Power: How People Make Decisions*, a methodology where experienced people are able to make quick decisions better than novices. First there are two critical processes:

1. Understanding how to size up the situation
2. Evaluating the action path forward (with your imagination)

Variation is the rollout of the RPD:

1. History – having done this before – the decision maker recognizes a pattern and knows how to react and act to it
2. Diagnostic – a pattern match is not exact, but there is a single historical pattern that can be adapted; or it maps onto multiple historical situations where there are connections – the decision maker adopts the closest pattern to the current situation
3. Evaluation– there is no match, understand the critical aspects to reason out the best path forward

While the RPD model sounds obvious, it's different from earlier decision theories.

An alternative approach is the Rational Choice Strategy where each perspective is evaluated, weighted, and ranked – the highest ranked provides the path forward.

RPD vs. Rational Choice

- When you cannot trust someone to make a critical decision, why do you trust them on the non-cortical decisions – use RPD
- If time is short or information is lacking – use RPD
- If you have no experience in the situation – use RPD
- If you have experience – go with which models feels best
- If it is a team decision, where all must make a commitment and adjust to multiple perspectives to increase the chance for success in the path forward – use rational choice strategy

Your reputation is all that you have. Your character is formed by how you show:

- Caring – do you display kindness towards others

- Citizenship – do you have compassion for others who are in common situation, environment, and circumstances that you find yourself a part? Compassion is the emotional bond that connects with those in your society
- Fairness – do you treat others impartially without favoritism or discrimination?
- Respect – do others hold a deep feeling of admiration of you based upon your behavior, abilities, qualities, or achievements?
- Responsibility – how accountable are you?
- Trustworthiness – how reliable are you?
- Honesty – truthful, factual, faithful, moral, accurate, open, genuine, just, real, honorable, straightforward, sincere, dependable, reliable
- Firm – do you exhibit steadiness without being excessive in power, action, attitude, determination, and strength of character?

And lastly, how do you take your personality, and make it work successfully in a team? What do you need to learn in addition to the technical parts of Junior Design to make your Junior Design team and project successful? How you will apply this reflection and learning going forward?

References:

1. Thompson C., "Full Tilt: When 100% of Cars Are Autonomous", *The New York Times*, November 8, 2017.
2. Hawkins, A.J., "Waymo is first to put fully self-driving cars on US roads without a safety driver - Going Level 4 in Arizona", The Verge, November 7, 2017, https://www.theverge.com/2017/11/7/16615290/waymo-self-driving-safety-driver-chandler-autonomous
3. Hevelke, A., & Nida-Rümelin, J. (2015). Responsibility for Crashes of Autonomous Vehicles: An Ethical Analysis. *Science and Engineering Ethics, 21*(3), 619-630.
4. Judith Jarvis Thomson. (1985). The Trolley Problem. *Yale Law Journal, 94*, 1395-1855.
5. Asimov, I. (1950). *I, Robot*. Greenwich, Conn: Fawcett Publications.
6. Ashley Halsey III, "NTSB says driver in fatal Tesla crash was overreliant on the car's 'Autopilot' system", September 12, 2017, *The Washington Post*, https://www.washingtonpost.com/local/trafficandcommuting/ntsb-says-driver-in-fatal-tesla-crash-was-overreliant-on-the-cars-autopilot-system/2017/09/12/38e5f130-9730-11e7-82e4-f1076f6d6152_story.html?utm_term=.90c472ced9d2
7. Fagnant, & Kockelman. (2015). Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transportation Research Part A, 77*, 167-181.
8. Buehler, Martin, Iagnemma, Karl, & Singh, Sanjiv. (2009). *The DARPA Urban Challenge: Autonomous vehicles in city traffic* (Springer tracts in advanced robotics; v. 56). Berlin: Springer.

9. Bonnefon, J., Shariff, A., & Rahwan, I. (2016). The social dilemma of autonomous vehicles. *Science (New York, N.Y.), 352*(6293), 1573-6.
10. Abraham, H., Lee, C., Brady, S., Fitzgerald, C., Mehler, B., Reimer, B., Coughlin J.F. "Autonomous Vehicles, Trust, and Driving Alternatives: A
11. Survey of consumer preferences", *MIT AgeLab & New England University Transportation Center*, White Paper, 2016.
12. Taub, E. A., "Envisioning the Car of the Future as a Living Room on Wheels", *The New York Times*, June 15, 2017.
13. Zhao, H.,Dimovitz, K., Staveland, B., Medsker, L., "Responding to Challenges in the Design of Moral Autonomous Vehicles", *The 2016 AAAI Fall Symposium Series: Cognitive Assistance in Government and Public Sector Applications Technical Report* FS-16-02, 2016.
14. Millard-Ball, A. (2017). Pedestrians, Autonomous Vehicles, and Cities. *Journal of Planning Education and Research,* 0739456X1667567.
15. Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H., & Didoff, J. (2016). The Grand Cooperative Driving Challenge 2016: Boosting the introduction of cooperative automated vehicles. *Wireless Communications, IEEE, 23*(4), 146-152.
16. Benderius, Ola, Berger, Christian, & Malmsten Lundgren, Victor. (2017). The Best Rated Human-Machine Interface Design for Autonomous Vehicles in the 2016 Grand Cooperative Driving Challenge. *Ieee Transactions On Intelligent Transportation Systems (Print),* IEEE transactions on intelligent transportation systems (Print), 2017.
17. Meyer, Gereon, & Beiker, Sven. (2014). "*Road vehicle automation*" (Lecture notes in mobility).
18. Hörl, S., Ciari, F., Axhausen, K.W., "Recent perspectives on the impact of autonomous vehicles", Working paper 10XX, *Institute for Transport Planning and Systems*, September 2016.
19. Bagloee, S., Tavana, A., Asadi, M., & Oliver, M. (2016). Autonomous vehicles: Challenges, opportunities, and future implications for transportation policies. *Journal of Modern Transportation, 24*(4), 284-303.
20. Will your self-driving car be programmed to kill you? (2015, June 04). *University Wire*, p. University Wire, Jun 4, 2015.
21. Patents; Researchers Submit Patent Application, "Methods for Operation of Autonomous Vehicles in Special Control Zones", for Approval (USPTO 20150370255). (2016, January 11). *Journal of Engineering*, p. 10714.
22. "Why Self-Driving Cars Must Be Programmed to Kill, A View from Emerging Technology from the arXiv", MIT Technology Review, October 22, 2015, https://www.technologyreview.com/s/542626/why-self-driving-cars-must-be-programmed-to-kill/ .
23. Lingard, R.W., (2010). Teaching and Assessing Teamwork Skills in Engineering and Computer Science. Journal of Systemics, 8(1), 34-37.
24. Brinkman, R., Kirschner, R., *Dealing with People You Can't Stand: How to Bring Out the Best in People at Their Worst*, McGraw-Hill Education, 2002.
25. Bramson, R. M. (1981). *Coping with difficult people*. Garden City, N.Y: Anchor Press/Doubleday.

26. Topchik, Gary S. (2007). *The first-time manager's guide to team building* (Safari Books Online). New York: AMACOM.
27. Moussa, M., Boyer, M., & Newberry, D. (2016). *Committed teams: Three steps to inspiring passion and performance*.
28. Pritchard, K., & Eliot, J. (2012). *Help the helper: Building a culture of extreme teamwork*. New York, N.Y: Portfolio/Penguin.
29. Murphy, J. J. (2016). *Pulling together: 10 rules for high performance teamwork*.
30. Ritter, A. (2010). *The 100/0 principle: The secret of great relationships*. Naperville, Ill: Simple Truths, LLC. [Note: the secret is "You take full responsibility (the 100) for the relationship, expecting nothing (the 0) in return."]