Thesis Project Portfolio

Turning Vehicle Controls Into Drive-By-Wire

(Technical Report)

Unsafe Streets: Are Automated Vehicles the Solution?

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science University of Virginia • Charlottesville, Virginia

> In Fulfillment of the Requirements for the Degree Bachelor of Science, School of Engineering

> > **Matthew William Deaton**

Spring, 2022 Department of Mechanical Engineering

Table of Contents

Sociotechnical Synthesis

Turning Vehicle Controls Into Drive-By-Wire

Unsafe Streets: Are Automated Vehicles the Solution?

Prospectus

Sociotechnical Synthesis

Anyone who grew up watching the Jetsons was in disbelief when they saw all of the technology that George Jetson had at his fingertips. One of the inventions that caught the eye of many was the video communicators that were used throughout the series. No one thought that in their lifetime would technology superior to theirs would exist. Then came video calls, and on phones the size of pocketbook. Another technological marvel is taking place, only this time it has to do with vehicles. Cars that drive themselves aren't just being researched, they are being tested across the country. They are on the cusp of being distributed to the general public.

However, technology doesn't happen in a vacuum, there are ethical questions that have to be considered with the arrival of automated vehicles. The Societotechnical paper will investigate the many repercussions and effects that this new technology will have on society. The SCOT framework was chosen to do this.

The Technical paper will discuss the steps taken to begin to make a 2008 Ford Escape self-driving. The first step, which was completed by this years team was to make the vehicle Drive-by-Wire. The end result involved making the braking, steering and throttle fully controlled by a remote.

Unsafe Streets: Are Automated Vehicles the Solution?

A Research Paper Submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science University of Virginia • Charlottesville, Virginia

> In Partial Fulfillment of the Requirements for the Degree Bachelor of Science, School of Engineering

Matthew William Deaton Spring 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor S. Travis Elliott, Department of Engineering and Society

Introduction of STS Topic

Self-driving cars, the phrase used to sound like something out of a science fiction novel, part of a far-off time that maybe future generations would see. However, with the rate technology has been improving, their advent is fast approaching. In the past 20 years, Automated Vehicles (AV's) have gone from barely being able to drive 8 miles in a desert to pursing legalization for for-profit use on the streets of San Francisco (Hawkins, 2022). If AV's are legalized they will impact society in many ways. This STS paper will use the Societal Construction of Technology (SCOT) model of analysis to deconstruct the societal impact of this developing technology. It will analyze the viewpoints of producers of the technology, advocacy groups concerned with the ramifications of the technology, and the apprehensions of some governmental groups. The concerns and points made by each group are important and need to be considered by society as a whole.

Introduction of STS Framework

The SCOT framework has two stages in its assessment of the social acceptance of technology. The first stage is Interpretive Flexibility, which involves understanding the people groups the new technology effects, and how they view these effects. These groups will hereafter be referred to as "stakeholders." Then, once these stakeholders and their different views have been identified, the SCOT framework explores how their views conflict. The second stage of SCOT is considering Closure, or the societal acceptance of the technology being investigated. This requires looking at possible permanent and temporary solutions to the problems of each social group.

Links Between STS Framework and AV's

The SCOT framework was chosen because the theory suggests that both technological development and how new technologies are used are driven by society. This means that these groups and their views will influence whether self-driving cars will be released to the streets of the United States at large. SCOT also states that how AV technologies are implemented, if they ever are, will also be shaped these by stakeholders.

Historical Context

There were over 38,000 crash deaths in in the U.S. in 2020 (DOT, 2020). The makers of AV's claim that their technology will reduce this number. It is possible that with further development their claim is correct. However, there is some controversy, especially after an automated Uber killed a pedestrian, Elaine Herzberg, in 2018. (Wired n.d.; Hawkins, 2020).

Analysis by STS Framework: Interpretive Flexibility

The stakeholders mentioned in this paper can be divided into three groups: first, developers of self-driving cars (Waymo, n.d.; Aurora n.d.), second, those who have concerns with the implementations or potential dangers of the technology (Chowdhury, 2020; Canis 2018; Alessandrini, 2021), and third, advocacy groups for or against the adoption of self-driving cars into everyday life. (CFLIC, 2022; Roy, n.d.).

This paper will look at two AV producers, Waymo and Aurora. Waymo was started in 2009 by Google (Waymo n.d.), Aurora was founded in 2017 (Aurora n.d.). They are two of the most promising AV companies. However, they are not the only companies investing in this technology. These companies use systems of lidar, radar, cameras, programs, and computers to create self-driving cars. The technology is still being tested, but Waymo and Aurora's big push for automated technology is that their systems won't get tired, or drunk, or decide to show off in

front of the attractive computer program driving next to it. Both companies have spent thousands of hours refining their programs (Waymo n.d.; Aurora n.d.). Waymo is trying to enter the taxi service specifically and has made substantial headway. The company has logged significantly more hours driven than any other over the last year. It has been taxiing its "trusted testers" around San Francisco and has been doing so since August 2020. The city of San Francisco recently granted Waymo the ability to start charging its riders, provided that there is a human "safety driver behind the wheel while the car is in operation" (Hawkins, 2022).

Aurora has made partners with two major trucking companies, Paccar and Volvo, and is attempting to break into the trucking industry (Aurora, n.d.). Aurora claims its driver is superior when compared to others due to the ability of their AV's to share information. Using this shared information, their cars create maps that are shared globally, and they claim the system will be extremely effective. However, the companies have yet to release their technology to the public at large.

This linking of maps and constant communication between cars is not viewed positively by some. Many have safety concerns. The opening of a vehicles navigation and driving systems to the internet presents a new avenue for those with malicious intent. Whenever anything is connected to the internet there is the potential for it to be hacked. A survey was conducted by the Institute of Electrical and Electronic Engineers, those who were surveyed indicated that many potential buyers wanted some kind of assurance that the owners of the vehicles would be in constant control (Chowdhury, 2020).

The California Foundation for Independent Living Centers or CFILC, is an advocacy group for those who are disabled. They want AV's to be developed so that people who are disabled, like the blind or those who suffer from severe cerebral palsy, can be more independent.

They advocate for the allowance of truly self-driving vehicles so that those who are unable to get a driver's license are able to get around. In the future, the AV's developed by Waymo and Aurora could accommodate these requirements.

Some critics of AVs oppose them on civil rights grounds. Alex Roy, author of the Human Driving Manifesto, claims that AVs threaten to sacrifice personal liberty in the name of safety. He argues that "the very language of self-driving is slavery" (Roy 2018). Roy argues that the advent of AV's will eventually lead to the removal of citizen's right to drive. Roy and those who support him claim that the human ability to make decisions is more important than safety. The Human Driving Manifesto is hoping to create an additional Constitutional Amendment, adding a "Right to Drive" alongside the other rights that are guaranteed to U.S. citizens. The Human Driving Manifesto's end goal is to use technology to aid human drivers, but not take over driving entirely. The technology required to truly have self-driving cars is still many years out. It may not be possible. Companies like Waymo and Aurora are using computer systems that rely on statistical data gathered from billions of hours spent testing (Waymo, n.d.; Aurora, n.d.). The problem with this approach is that statistical data while often very accurate, cannot account for every variable in the same way a human does. Roy and his supporters are acting preemptively hoping to answer the question of whether there is a right to drive before it is even asked.

There was a workshop held in 2021 by many Infrastructure Owners and Operators (IOOs) to discuss the role of infrastructure in the event of United States-wide implementation of AVs. Many possible problems were identified by the workshop. One was an anticipated increased load on roads from AV platooning. This is the linking of several AV's together and setting them to drive to the same place. Another was the need to improve pavement quality to reduce interference with AV's programs. There were so many others that a majority opinion held

by IOOs was that more data is required because there are so many potential problems posed by AVs (Alessandrini, 2021).

Federal and state governments make up another set of stakeholders. Lawmakers are having difficulty legislating AV use. Entire new suites of bills and laws are being written to prepare for their possible implementation (Canis 2018). It is very likely that if AV's are successful and gain popularity that large portions of the United States infrastructure system will have to change to accommodate the problems identified in the IOO workshop. The cost of a possible complete retooling of the nations infrastructure may not be worth the convenience of self-driving cares.

In 2004 the Defense Advanced Research Projects Agency aka DARPA, issued a "Grand Challenge." DARPA offered a reward to any team that could create an autonomous vehicle that could navigate their 150-mile course through the Mojave Desert. No team received the reward, in fact the team that made it the furthest went 7.32 miles. The next year, undaunted, DARPA reissued the challenge. Every team save one passed the previous year's record, and 5 finished the race entirely (Russel, 2006). If DARPA's Grand Challenge is any indication, it is clear that progress is steady. It is also clear the technology is here to stay. On today's roads, many current operating systems already have minor forms of self-driving capability i.e., lane assist, brake assist, smart cruise control, self-parallel parking systems etc. are available in most cars. The technology is also prevalent outside the automotive industry: robotic manufacturing, autopilot in airplanes, robotic cleaning systems etc. The list of automated technologies is endless, and it only keeps growing. The technology being created by AV companies has clear benefits, but it is unknown as to whether it will fulfill its intended purpose of being truly independent of human oversight.

Interpretive Flexibility: Problems and Conflicts

The problems between the advocacy groups like CFLIC and the Human Driving Manifest are obvious. The disabled people CFLIC are advocating for cannot drive and those at the Manifest don't believe that machines can ever truly make the decisions that humans can. However, there are other less obvious issues that are just as important. If Aurora is successful in its implementation of automated trucking, it will end the truck drivers way of life. Thousands of jobs will be lost. The insurance industry will also have to figure out how to insure the vehicles. The question of who gets sued when there is an accident will have to be answered. Each state will have to decide whether they will allow self-driving cars on their roads. It's also possible that the IOO's worst fears are confirmed and every road in the US will have to be repaved and every bridge will have to be reinforced.

Closure

The problems posed by these groups are not without solutions. Some of the solutions will be more difficult than others but acceptable compromises are possible.

With regards to Mr. Roy and his compatriots, driving isn't legally a right, it is a privilege. However, laws could be made to ensure that humans will always be allowed to drive their own vehicles should they so choose.

Instead of hiring truckers to drive, they could be hired as security for the trucks. Trucks can carry large amounts of expensive equipment and it would be unwise to leave them unattended for long trips across the country. The likelihood of a raid on the vehicle or an unauthorized detour is worth the cost of hiring someone to watch the vehicle as it travels to its destination.

If certain states decided that they do not want AV's in their borders, hybrid vehicles could be made to ensure that they are able to be driven everywhere. Indicators could be put on the vehicles that show when the car is driving itself.

If lawmakers decide that the owner of a vehicle is responsible for a crash, then the laws don't change. In this case the function of the vehicle's automation would be similar to autopilot for airplanes. If lawmakers decide that a company is at fault for its program, regardless of whether it failed or performed perfectly, a lawsuit cap could be made so that the companies are not bankrupted.

Personally, I am not an advocate for the companies being held universally responsible for crashes. I think that it is important for people to have responsibility for their property and actions. I believe that if laws require the car companies to be held universally liable, they create a culture that habitually pawns off responsibility at every turn...no pun intended.

Conclusion

Regardless of my personal opinions, AV technology, and the research leading to it have done many good things. It has saved countless individuals who lack the inborn skill to parallel park. It has benefited many drivers without being fully realized, and as of right now has not demonstrated itself to be a threat to anyone's liberties. At this current time, it is unclear if the technology will be universally implemented but given the current progress it is not unrealistic to be hopeful.

References

- Chowdhury, A., Karmakar, G., Kamruzzaman, J., Jolfaei, A. and Das, R. "Attacks on Self-Driving Cars and Their Countermeasures: A Survey," in IEEE Access, vol. 8, pp. 207308-207342, 2020, doi: 10.1109/ACCESS.2020.3037705.
- Alessandrini, A., Domenichini, L., & Branzi, V. (2021). The Role of Infrastructure for a Safe Transition to Automated Driving. Amsterdam: Elsevier.
- Aurora. (n.d.). Our Company. Aurora.tech. https://aurora.tech/company
- Canis, B., & Library of Congress (issuing body) (2018). Issues in Autonomous Vehicle Deployment ([Library of Congress public ed.). Washington, D.C.: Congressional Research Service.
- CDC. (2016, July 6). Motor vehicle crash deaths. Centers for Disease Control and Prevention. https://www.cdc.gov/vitalsigns/motor-vehicle-safety/index.html.
- CDC. (2016, December 5). Motor Vehicle Injuries. Centers for Disease Control and Prevention. https://www.cdc.gov/winnablebattles/report/motor.html
- Census Bureau. (2021, October 31). Population Clock. Population Clock. https://www.census.gov/popclock/
- USDOT. (2021, May). Early Estimate of Motor Vehicle Traffic Fatalities in 2020. United States Department of Transportation. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813115
- Federal Highway Administration. (2020, February 28). Highway Statistics 2018. Table DL-22 -Highway Statistics 2018 - Policy | Federal Highway Administration. https://www.fhwa.dot.gov/policyinformation/statistics/2018/dl22.cfm
- Hawkins, A. J. (2020, December 7). Uber's fraught and deadly pursuit of self-driving cars is over. The Verge. https://www.theverge.com/2020/12/7/22158745/uber-selling-autonomous-vehicle-business-aurora-innovation
- Marshall, A. (2020) Uber Gives Up on the Self-Driving Dream. Wired. https://www.wired.com/story/uber-gives-up-self-driving-dream/
- NHTSA. (2021, June 3). 2020 Fatality Data Show Increased Traffic Fatalities During Pandemic. https://www.nhtsa.gov/press-releases/2020-fatality-data-show-increased-traffic-fatalitiesduring-pandemic

Roy, A. (2018, March 5). This Is the Human Driving Manifesto. The Drive. https://www.thedrive.com/article/18952/this-is-the-human-driving-manifesto

Toups, D. (2021, January 6). How Many Times Will You Crash Your Car? Forbes. https://www.forbes.com/sites/moneybuilder/2011/07/27/how-many-times-will-you-crashyour-car/?sh=1476e6424e62

Waymo. (n.d.). October 7, 2021 https://waymo.com/company/#story.

- Hawkins, A. J. (2022, March 1). *Waymo to start charging money for its robotaxi rides in San Francisco*. The Verge. Retrieved May 11, 2022, from https://www.theverge.com/2022/3/1/22956335/waymo-robotaxi-rides-paid-san-franciscocpuc
- CFLIC. CFILC Advocacy: Support Development of Self-Driving Cars! / California Foundation for Independent Living Centers. (2022). CFLIC. Retrieved November 5, 2022, from https://cfilc.org/issues/self-drivingcars.php?msclkid=4f5e1b4dcff211ec97f63ff2640d4ed8
- Russel, S. (2006, January). *DARPA Grand Challenge Winner: Stanley the Robot! Popular Mechanics*. Popularmechanics.Com. Retrieved May 11, 2022, from https://web.archive.org/web/20071029171215/http://www.popularmechanics.com/science /robotics/2169012.html?page=3

ME DESIGN TECHNICAL REPORT

THE TRANSFORMATION OF VEHICLE CONTROLS TO DRIVE-BY-WIRE

Teammates

Henry Goodman

Alex Pascocello

Jacob Deane

Vishal Singh

Logan Montgomery

Matthew Deaton

University of Virginia

5/09/2022

TABLE OF CONTENTS

- 1. Introduction
- 2. Background and Essential Knowledge
- 3. Design Process
- 4. Final Design
 - 4.1. Full Design Diagrams
 - 4.2. Throttle-by-Wire System
 - 4.3. Brake-by-Wire System
 - 4.4. Steer-by-Wire System
 - 4.5. Emergency Stop and Autonomation Switch
 - 4.6. Sensor Suite
 - 4.7. Software
- 5. Mathematical and Numerical Analyses
 - 5.1. Motion Model Comparison
 - 5.2. PID Tuning
- 6. Experimental Validation
- 7. Operations Manual
- 8. Conclusions and Future Work

1 INTRODUCTION

Autonomous vehicles are revolutionizing the transportation industry. If implemented correctly they can provide greater freedoms to the eldery and disabled, create safer roads, and give back time to commuting workers. While this technology is not ready to be assimilated into society, there are steps being taken to introduce semi-autonomous vehicles to the road. There is a scale of autonomy for such vehicles that ranges from zero, no autonomy, to five, full autonomy, as seen in Figure 1. Currently, some electric vehicles already use low levels of automation such as self-parking and lane assist (Shipley, 2021). This would not be possible without the ability to control the vehicle's integral driving functions such as steering, throttle, and braking. The goal of this project is to lay the groundwork for the future autonomous control of a Ford Escape by creating an electromechanical drive-by-wire system to externally steer the vehicle. This will allow the implementation of semi-autonomous processes and eventually full autonomy.



Fig. 1. Levels of Autonomy as Defined by The Society of Automotive Engineers (Wevolver, 2020)

Early on in the design process, the team established a set of target specifications that reflected the needs of the prospective clients. This target specification list can be seen below in Figure 2. In the past, there have been multiple attempts to create a higher level autonomous vehicle with some of these design elements, with varied levels of success. The Stanford Stanley project is a modified version of a Volkswagen Touareg R5, designed for competing in DARPA Grand Challenge in 2005. Stanford's Stanley placed first in this competition, and boasted a completion time of 6 hours and 54 minutes (see Appendix F). While this vehicle included important components of the target specifications mentioned above such as functional steering, throttle, and braking systems, it lacks important systems, such as the integration of a remote control, allowing the car to be manually controlled from an outside user.

Outside of vehicles designed for the DARPA Grand Challenge, there are multiple companies currently developing level 4 autonomous vehicles. Waymo, formerly the Google Self-Driving Car Project, has developed a fleet of functioning autonomous vehicles, operating in Phoenix, Arizona (Schwall et al., 2020). While these vehicles are impressive and present groundbreaking breakthroughs in autonomous driving technology, they operate on a slightly different principle than what has been defined for our project by the initial target specifications. Waymo vehicles operate off of a highly detailed map that is premade for a specific city in conjunction with the sensor suite integrated into the vehicle (Laukkonen, 2020). For the purposes of this project, the ultimate goal is to create a vehicle that does not need to operate in a specific city, or be limited to use only on well maintained and marked roads. There is definitely much to take away from the way Waymo's autonomous vehicles operate, but the end goal of this project differs categorically from Waymo's Self-Driving Car.

The main objectives of this project are to create a vehicle that is able to map its surroundings in real time and make decisions regarding what actions to take in order to safely drive to its objective. This process should not be restricted to operating solely on paved city streets, and as a result cannot be solely reliant on external mapping and road markings. A full list of these goals can be seen below in Figure 2, and potential design solutions based around the completion of these goals can be seen in Figure 3. These goals are not expected to be completed in their entirety by the end of the year, and are more reflective of the long term, multi-year goals of the project. For this year, the main objectives of the project are to design and integrate functioning steer-by-wire, throttle-by-wire, and brake-by-wire systems into the car, as well as create a design for the essential sensors for progress in future years.

2 BACKGROUND AND ESSENTIAL KNOWLEDGE

The Ford Escape being used for the team's project came from a previous Virginia Tech project with similar goals. When the car was received it was stripped of most subsystem connections including all LiDar and radar systems. There was a mounted pulley with a rusted stepper motor already installed for the brake-by-wire system, but it had to be replaced due to rust. The computer's hard drive became corrupted and had to be replaced, but some past Arduino and Python code was salvageable.

Beside the state of the car, no report or documentation is known regarding the success of the previous team. While the current TA involved with the project spearheaded the previous project at Virginia Tech, his involvement is purely educational, helping guide the team with his previous knowledge of the systems.

There were several required subjects the team researched in preparation for the drive-by-wire project. These topics included electrical systems of electric power steering and throttle system components, closed-loop control systems specializing in PID controllers, and Controller Area Network (CAN) bus readings utilizing python codes. The technological background aided the team in accomplishing the set forth goals and creating functional steer-by-wire, brake-by-wire, and throttle-by-wire subsystems.

Advancements were made on previous drive-by-wire projects in several different ways in accordance with customer requirements. Firstly, the car will have an emergency stop feature that will immediately halt all autonomous capabilities and give control over to the driver. Furthermore, there is an accessible switch to easily switch from driver to autonomous mode in a non-emergency setting. A second advancement to the drive-by-wire vehicle is a PID feedback closed-loop control system for each subsystem to ensure a smooth and safe controlled driving experience. This control system will be used, for example, to confirm steering angles while driving at high speeds where the steering wheel could naturally return to a neutral position due to tire friction. In these ways the team's project improves on other drive-by-wire systems.

3 DESIGN PROCESS

To determine the goals of the project and design, we first surveyed a customer on what they would like to see in their drive-by-wire vehicle that would eventually become autonomous in the future. For the scope of this year's project goal, a strictly drive-by-wire car with no autonomy, the most important needs from the customer were a "drive-by-wire system with practical controller," and it "must have steering and override electrical or mechanical

capabilities," referring to the E-stop needed for the car. The biggest customer need was that the controller was "safer than a human operator," which was emphasized in our target specifications.

Taking into account the relation to customer needs, the technical importance, and the difficulty to implement, target specifications were generated based on important safety and performance criteria for a drive-by-wire system and the car as a whole. The full Quality Function Deployment (QFD) chart is shown in the appendix, but a short summary is shown in Figure 2 below.

Turning resolution	0 - 60mph acceleration time	Response time for Braking	Response time for throttle	Obstacle avoidance distance	Response time for steering	Total braking distance from 60- <u>Omph</u>	Throttle (speed changing) resolution					
\checkmark	\downarrow	\checkmark	\checkmark	0	\checkmark	0	\checkmark					
	Marginal Target Value											
3 deg	10 s	0.5 s	0.5 s	5 ft	0.5 s	200 ft	3 mph					
	Т	echnical Importar	nce (without takin	g into account	difficulty)							
10.4%	11.2%	16.4%	9.0%	9.6%	13.7%	16.4%	13.4%					
		*				*						

Fig. 2. Summary of the QFD Chart

The two most prioritized specifications are the total braking distance from 60-0 mph and the response time for braking. The target specifications for response time for braking and throttle were the focus of our testing, which will be explained in the mathematical and numerical analyses section.

To brainstorm and decide potential solutions to implement drive-by-wire in the braking, throttle, and steering, concept generation was used. The full concept generation, including the

functional decomposition, will be found in the appendix, but Figure 3 shows below a summary of the different solutions brainstormed and which solutions will be implemented.

	Design Solutions															
			Brak	ing				Thrott	le		Steering					
Solutions	Actuator	Power	Input Control	Microcontrol ler	Feedback Control Method	Method	Power	Input Control	Microcontrol ler	Feedback Control Method	Method	Power	Input Control	Microcontrol ler	Feedback Control Method	
1	Stepper Motor	12V Car Battery	GUI on Onboard PC	Arduino Mega/Uno	CAN Bus Data	DigiPot Signal	12V Car Battery	GUI on Onboard PC	Arduino Mega/Uno	Pre-Existing Imbedded Potentiomet er Suite	Torque Sensor Message to ECU	12V Car Battery	GUI on Onboard PC	Arduino Mega/Uno	CAN Bus Steering Position Data	
2	Hack Power Braking	12V Car Battery	GUI on Onboard PC	TI Microcontrol Ier	Absolute Encoder	Servo Motor to Push Pedal	12V Car Battery	GUI on Onboard PC	TI Microcontrol Ier	Quadrature Encoder	Torque Sensor Message to ECU	12V Car Battery	GUI on Onboard PC	TI Microcontrol Ier	CAN Bus Steering Position Data	
3	Stepper Motor	12V Car Battery	GUI on Onboard PC	Parallax P2 Chip	Sensor Suite to Measure Speed	Pneumati c Actuator to Push Pedal	12V Car Battery	GUI on Onboard PC	Parallax P2 Chip	Pre-Existing Imbedded Potentiomet er Suite	Motor in Line With Steering Column	12V Car Battery	GUI on Onboard PC	Parralax P2 Chip	CAN Bus Steering Position Data	

Fig. 3. Concept Generation Design Solutions

Based on the morphological analysis, three potential solutions were created, and then these designs were evaluated based on their strengths and weaknesses. Out of three designs, solution number 1 was selected, as denoted by the yellow highlighted cells.

Our potential design was then compared to other similar designs by other institutions or existing solutions in the industry in a concept selection process. Figures 4 and 5 show the concept selection screening and scoring processes.

Potential Solution →	CMU's Boss	Stanford's Stanloy	VT's VistorTango	MIT's Tales		NuTonomy Drivorlass Taxi	Our col 1	Our col 2	Our col 2
Selection Criteria 🗸	CIVIU S BOSS	Staniord's Stanley	VI S VICtor lango	IVIT S TAIOS	VOIVO Drive Ivie	Nutonomy Driveness Taxi	Our sol. 1	Our sol. 2	Our sol. 5
Turning Resolution	+	+	-	+	0	+	+	+	+
0-60mph Acceleration	+	-	-	-	-	-	+	+	0
Response Time for Braking	0	-	-	-	-	-	0	-	-
Response Time for Throttle	+	+	+	+	+	+	+	+	+
Obstacle Avoidance Distance	0	-	-	-	-	-	0	-	-
Response Time for Steering	0	-	-	-	-	0	+	+	+
Total Braking Distance From 0-60mph	+	+	+	+	+	0	+	-	0
Throttle Resolution (Speed Change)	+	+	+	+	0	-	+	+	+
Sum +'s	5	4	3	4	2	2	6	5	4
Sum 0's	3	0	0	0	2	2	2	0	2
Sum -'s	0	4	5	4	4	4	0	3	2
Net Score	5	0	-2	0	-2	-2	6	2	2
Rank	2	5	7	5	7	7	1	3	3
Continue?	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes

Fig. 4. Concept Selection (Screening)

Inferences on how the different subsystems and target specifications of other solutions compared to our solutions were made based on performance, type of car, and the types of DARPA challenges they competed in. Our primary solution ranked the best in this qualitative analysis.

Solution →		CMU	s Boss	Stanfor	d's Stanley	VT's V	ictorTango	MIT	s Talos	Our	sol. 1	Our	sol. 2	Our	sol. 3
# Selection Criteria ↓	Weight	Rating	Weight	Rating	Weight	Rating	Weight	Rating	Weight	Rating	Weight	Rating	Weight	Rating	Weight
1 Turning Resolution	10%	5	0.5	5	0.5	0	0	5	0.5	5	0.5	4	0.4	5	0.5
2 0-60mph Acceleration	10%	5	0.5	1	0.1	1	0.1	1	0.1	5	0.5	5	0.5	3	0.3
3 Response Time for Braking	20%	1	0.2	1	0.2	1	0.2	3	0.6	0	0	1	0.2	1	0.2
4 Response Time for Throttle	10%	5	0.5	4	0.4	5	0.5	4	0.4	5	0.5	5	0.5	5	0.5
5 Obstacle Avoidance Distance	10%	1	0.1	1	0.1	1	0.1	1	0.1	0	0	1	0.1	1	0.1
6 Response Time for Steering	10%	1	0.1	1	0.1	1	0.1	1	0.1	5	0.5	5	0.5	5	0.5
7 Total Braking Distance From 0-60mph	20%	4	0.8	4	0.8	4	0.8	4	0.8	5	1	2	0.4	3	0.6
8 Throttle Resolution (Speed Change)	10%	4	0.4	5	0.5	4	0.4	4	0.4	4	0.4	4	0.4	4	0.4
	100%		3.1		2.7		2.2		3		3.4		3		3.1
		RANK	4		2		1		3		5		1		4

Fig. 5. Concept Selection (Scoring)

In the concept scoring process, relative weights were applied to each selection criteria based on importance determined during the target specifications process. The rankings aligned with the concept screening rankings, with our solution 1 ranked the highest by receiving top scores in all categories besides response time for braking and obstacle avoidance distance.

4 FINAL DESIGN

4.1 FULL DESIGN DIAGRAMS





Fig. 6. Block System Diagram

Fig. 7. Complete Wiring Diagram

4.2 THROTTLE-BY-WIRE SYSTEM

The design for the throttle control system is a direct electrical approach. Instead of using an actuator to pull the pedal (as is the case for braking) the team thought a less obtrusive and more direct option would be to mimic the signals that the throttle pedal sends the electrical control unit (ECU) to tell the car to tell the engine to accelerate. Therefore, the problem could be solved solely by using digital-to-analog converters (DACs). The throttle pedal has three potentiometers that send different ranges of voltages to the ECU to be interpreted as a throttle amount. These three signals need to be hijacked and mimicked. For this we use three DACs controlled by an Arduino UNO. The relevant electrical components are highlighted in Figure 8.



Fig. 8. Throttle Electrical Diagram

Our throttle code tells the DACs exactly what voltages to output in order to achieve a specific throttle percentage from 1-100. The three throttle signals each have their own unique idle voltages and voltage ranges that the ECU then uses to compute a throttle percentage. These are depicted in Figure 9.



Fig. 9. Throttle signals sent to ECU for idle and throttle pedal fully pressed

These values were discovered experimentally by measuring the signals directly. Our code to control throttle takes these ranges and applies a linear distribution to the outputs from our DACs. So, the percent of throttle requires the three correct voltage values to be sent from each DAC's output channel.

```
void throttle(float thr_percent) {
  float accelerator_command = 1800 * (thr_percent / 100);
  dac1.setVoltage(3730 - (1.3*accelerator_command), false);
  dac2.setVoltage(1320 + accelerator_command, false);
  dac3.setVoltage(815 + accelerator_command, false);
}
```

Fig. 10. Throttle Signal Mimic Function (in bits)

A three channel relay is able to switch the car between autonomous mode and manual mode by switching the voltages that the ECU receives from our DAC outputs to the throttle pedal outputs. In order to ensure accurate and safe acceleration, this system will eventually be integrated with and controlled by a PID controller. A feedback loop with the throttle messages from the car's CAN bus will allow us to make sure that the car is autonomously throttling as expected even when changing slopes.

4.3 BRAKE-BY-WIRE SYSTEM

The braking subsystem is the only subsystem controlled by external actuation rather than internal signal mimicking. The team uses a stepper motor driven by a DM860T stepper driver and controlled by arduino code through ROS via our operator.py script. A power board inverts 120V AC to 32V DC in order to power the motor. The relays for braking switch the power on and off to the motor, so no external actuation is possible without them flipped. This system features a PID control loop using the brake's CAN bus status to ensure that the brake reaches the correct position in the most efficient and smooth manner with little oscillation. This is motivated by error factors such as the lack of precise positioning with the stepper motor and slack in the pulley cable.

The CAN bus feedback is also used in the zeroing function for braking. As mentioned, the stepper motor does not record position. This means that if the motor slips, the wire connecting it to the brakes may become taut and the brakes may become compromised. To ensure there is no slack in the pulley when the car is put into autonomous mode, the brake zeroing function spins the stepper motor until the CAN bus registers that the brake is pressed. Once this occurs, the function slowly steps back the motor until the CAN bus registers the brakes as released. This zeroing function is imperative to ensuring a taut cable and immediate braking. The relevant electrical components are highlighted in Figure 11 as well as CAD models of the pulley system.





Fig. 11. Electrical (right) and physical (left) diagram of braking components

4.4 STEER-BY-WIRE SYSTEM

Electric power steering uses a motor to assist driver steering when a torque is applied. The torque sensor sends two voltages ranging from zero to five volts. These voltages stay around 2.5V when no torque is being applied and then spike or drop by similar amounts when torque is applied. Understanding and recording these values is required to emulate the signals sent by the torque sensor. The steering control system's final design works quite similarly to the throttle control system. It uses a DAC to emulate the voltage signals for torque to the ECU, mimicking what the torque sensor in the steering column would give it to assist a steering motion from the user. The steering system uses a single dual channel DAC called the MCP4922 which can output two different voltages. These signals are sent to the ECU when the car is in autonomous mode and the relevant electrical components are highlighted in Figure 12 below.



Fig. 12. Electrical diagram (steering components)

The steering signals that are being recreated are torque signals, so they spike when torque is applied and then return to a zero torque value. There are two torque signals to be mimicked, as

opposed to throttle where there are three position signals. Figure 13 explains how the torque signals move when torque is applied.



Fig. 13. Steering signals sent to ECU for zero torque and a small experimental torque

Understanding how these messages work was required to write the subsequent control code. This code takes a single input of a negative or positive bit value to offset each signal by. This is the input that we will then use to map the left stick of the remote controller. Since this is giving a torque command, PID control is necessary to give the steering wheel positional control, so when the user lets go of the stick the steering wheel will return to its straight position.



Fig. 14. Steering Arduino Function

4.5 EMERGENCY STOP AND AUTONOMATION SWITCH

The car will have the ability to switch between autonomy and manual control via two methods of input on the center console: the emergency stop (E-stop) button and the autonomy switch. These are connected in series with the relay circuit, so that users will need to actuate both inputs to enter autonomous mode, but just one to disable it. All signals being sent to the ECU and the power for the stepper motor run through relays. Flipping these relays is the most direct way to return the car to manual operation, giving the user control over all subsystems. A future improved version of the emergency system might entail an immediate full-braking command or a pull over and stop sequence. These are some semi-autonomous processes to be further developed by future teams. A picture and diagram of the E-stop and autonomy switch are provided in Figures 15 and 16.



Fig. 15. Emergency stop inputs: autonomy switch (left), E-stop button (right)



Fig. 16. Diagram for autonomy relays and E-Stop inputs

4.6 SENSOR SUITE

The Ford Escape came with two Blackfly S USB3 FLIR cameras, a 64 point LiDAR system and two GPS units. The cameras are able to form point cloud images and will be used by

future teams to gather data to avoid obstacles, vehicles and pedestrians. The next step for the cameras is to create a point cloud system represented as matrices from each camera's feed. A python code can be used to do this. The 64 point LiDAR system was replaced with a 128 point system made by Ouster for more accurate recordings. The Ouster system uses a 24V power supply and connects to the main computer using a USB to Ethernet cable. The LiDAR sensor should remain unplugged when not in use, because if it remains plugged in it will continually scan its surroundings and wear itself out. The sensor communicates to the computer using IPv4. When the LiDAR is fully connected, open a terminal window and use the command: *ping -4 -c3 os-122148000210.local* to activate it.

If this command runs and a package is sent and received, this indicates the sensor is connected to the computer and is able to send and receive data. Once the sensor is connected it needs to be configured, and will need to be reconfigured whenever disconnected from the main power. The configure command is:

python3 -m ouster.sdk.examples.client os-122148000210.local configure-sensor

Once the sensor is configured, it is ready to collect and receive data using the command: *python3 -m ouster.sdk.examples.client os-122148000210.local live-plot-signal*

This will broadcast a black and white image that will use the live data from the sensor. The Ouster sensor data is able to be used by rvis and broadcast live data that will be used when the car is made autonomous. To do this first enter catkin_ws/src. Run the command: *source ../devel/setup.bash*

Then the command:

roslaunch outster_ros ouster.launch sensor_hostname:=10.5.5.50 udp_dest:=10.5.5.1 metadata:=metadata.json Open rvis in a new terminal, add a point cloud by topic and change the fixed frame to "os_sensor" this will pull up a live feed from rvis. Figure 17 shows the final imaging when the above prompts are initialized successfully.



Fig. 17. LiDAR imaging taken in reactor room where car is worked on

4.7 SOFTWARE

The software suite on the Escape is meant to combine all of the concepts above into one centralized computing environment. This was accomplished with the Robot Operating System (ROS) middleware suite, which handles the operation of programs and the communications between different programs in the suite. Doing so allows for quick and clear communication methods between Python and Arduino files, as well as an external joystick and the car's onboard CAN bus system.

The software suite can be broken up into the 3 groups of hardware they reside on: the CAN bus, the onboard desktop computer, and the Arduino Unos. The CAN bus is the car's

onboard computer that allows the car's many microcontrollers and devices to communicate. This can be read using a PCAN to USB adapter that translates the CAN bus' messages into hexadecimals that the computer can read. The onboard computer houses many of the programs that will be run for the operation of the car. This includes scripts such as the CanBusReader.py script, which will decode the CAN messages, the joystickControl.py script, which will take input from a user-controlled joystick, and the operator.py script, which will take the decoded CAN messages and joystick input and decide what messages to send to the Arduinos accordingly. Lastly, the Arduinos convert the messages from operator.py into voltages, which then are sent to each subsystem to operate them. Once these systems change, their CAN messages will as well, and operator.py compares the new CAN values with the joystick input to make sure everything is operating properly. This is done using PID tuning, and is present on both the braking and steering subsystems.

Using these feedback loops, every subsystem can be checked for accuracy. This is especially important with the stepper motor for the brake, because, unlike the electrical systems like the torque sensor for the steering or the potentiometers for throttle that are relatively easy to set, this is a physical system. This means it is much more prone to error, such as a bad step from the motor or a loose or slipped cable from the motor to the brake. If an event like this happens, it will be immediately clear, as the CAN bus value for braking will not match the input value, allowing the software to adjust the brake as necessary. A full software logic tree is provided in Figure 18.



Fig. 18. Software system diagram

5 MATHEMATICAL AND NUMERICAL ANALYSES

5.1 MOTION MODEL COMPARISON

In order to validate the car's motion systems, students decided to compare reactions to a computer motion model of the car. Luckily, the Compositional Systems Lab at the University of Arizona created a Gazebo plugin called "catvehicle" as a testbed for autonomy. The model used is a Ford Escape (shown in Figure 19). This plugin features a controllable car model complete with sensors. Both the motion model and the actual car were fed acceleration step inputs for throttle, and the velocity of each system was measured in response. Trials were run at with inputs

of 8% throttle, 10% throttle, and 12% throttle, and the reaction was plotted for easy comparison in Figures 20, 21 and 22 respectively.



Fig. 19. Catvehicle plugin on Gazebo



Fig. 20. Trial with 8% Throttle



Fig. 21. Trial with 10% Throttle



Fig. 22. Trial with 12% Throttle

Interestingly, the results showed that the real car has a smoother acceleration than the simulated catvehicle. Overall, the car performs very similarly to the motion model, however there is much more internal noise in the modeled system. The systems are expected to converge more closely toward at cruising speeds because external factors like static friction have been overcome and air friction has plateaued. This behavior is true of the 8% and 12% trials, however the two systems failed to converge on our 10% trial.

5.2 PID TUNING

In order to minimize the error in brake position and eliminate jitter from the overall motion, a PID controller was implemented. PID controllers require tuning to achieve the desired motion, meaning that our Kp, Ki, and Kd constants must be adjusted in the calculation of the output command. In order to tune our system, the brake controller was fed a constant step input of reaching a brake position of 124 from an unpressed position of 127. There are several methods of tuning a PID controller, but our team chose to follow the Ziegler-Nichols method. This consists of first finding a system's ultimate gain (Ku), or the Kp value that causes constant undamped oscillation. Several equations relate Ku and its period of oscillation to the suggested values for the constants. The relation used in our braking system can be found in figure 23 below.

PI a. $P = 0.45K_u = 18$ b. $I = 0.54K_u/T_u = 7.2$

Fig. 23. Ziegler-Nichols equations for a P and I term only

Results were recorded for a PID controller using just a proportional (Kp) term (Ki=Kd=0), and a proportional and integral term (Kd=0). The trial with all three terms led to a worse result than just P and I and, therefore, is not shown. Figure 24 shows the attempt of the system to reach a brake position of 124 with a Kp (left) and a Kp and Ki value (right) found from the Ziegler-Nichols method with a Ku of 40 and period of 3 seconds.



Fig. 24. PID Brake Test (124 position step input), Left: P only, Right: P & I only

The results show the improvement of the system dampening the oscillation after adding in the integration term. Only two oscillations before reaching the setpoint is fairly good, although further tuning may be beneficial for future teams to pursue.

6 EXPERIMENTAL VALIDATION

The braking and throttle systems were tested for response times in order to ensure that the target specifications have been met for the car to be "safer than a human operator" as previously quoted from the customer. Step inputs for a brake position of fully pressed and a throttle value of 25% were fed into both systems and the time of the response was measured in Figures 25 and 26.



Fig. 25. Braking test results

The black lines and dots above the data illustrate the two separate response times of the system. The first two dots represents the time between the input given to the system and the beginning of the system response. For braking, this response time was 0.13 seconds. The second span of dots illustrates the time for the system to reach its desired state. These two response times were summed to show the total response time from user input until system completion. The total braking response time was 0.7 seconds. In fact, the human response time (moving foot from throttle to brake) is 0.75 seconds, which makes the system safer than a human operator.



Fig. 26. Throttle test results

This test was repeated on the throttle system. The system response time was .72 seconds (user input until system reaction), and the total response time was 5.22 seconds (user input until reaching 25% throttle). Due to the nature of throttle response, it takes more time for the system to reach the value specified by the step input.

7 OPERATIONS MANUAL

As a division of VICTOR Lab, we have developed this operations manual in order to record the general operating procedures necessary for the safe handling of the drive-by-wire 2008 Ford Escape Hybrid outlined in the document. This document has been designed to outline fundamental structure, development, and deployment of this system.

Before operating the system, it is essential to ensure that the 12V battery in the front of the car is connected, and that the high voltage (HV) battery in the trunk of the car has the

disconnect switch turned to the locked position. The locked position will have the crossbar handle of the switch sitting horizontal, as can be seen by the lock arrow close to the switch.



Fig. 27. Hybrid battery disconnect switch diagram

For the purpose of this description, it will be assumed that the car's non-native electronics are plugged into wall power via the power strip in the back of the car. To switch over to battery power, first locate the strip, the computer, the power inverter (labeled 'Pure Sine Inverter'), the battery, and the switch in the back of the vehicle. Before unplugging the power strip, make sure the computer is properly shut down. After this, unplug the strip from the wall and plug it into the outlet on the side of the power inverter. The battery is connected to a pink plastic clip, which will connect to a matching red clip connected to the car that should be located in the back of the vehicle. Plug the battery's pink clip into this red clip. Lastly, turn the safety switch (small gray box) on using the red key zip-tied to the switch. Once this is complete, you should be able to turn

the inverter on using the small circular power button located next to the outlet, which will supply power to the system.



Fig. 28. From left to right, front to back: Power strip, computer monitor, joystick, power inverter, battery-kill switch, 12V battery, computer.

Next, make sure the Arduinos are connected to the car systems. There are 4 total pin connections that need to be plugged into the arduinos. The pin connector with the red, yellow, and green wires is for the braking system and should be connected to the arduino with a single connector. The connector with the blue, yellow, black, and green wires is the throttle connector and should be connected to the only 4-pin connector on the other arduino. The remaining two 3-pin connectors are both for steering. The one with a blue, purple, and black wire should be connected to the middle pin connector. The last connector has blue, purple, and red wires and should be connected to the final left 3-pin connector. Finally, the two blue cables attached to extenders from the computer should both be connected to the arduinos.



Fig. 29. Pin connector locations

Make sure the CAN bus reader (white cable with a small box on it and a red light) is plugged into the car near the driver seat and the computer as well. The system should now be independently operable.

All software needed for the operation of the vehicle is located in the *catkin_ws/src/car_chassis_controller_s22* directory of the computer. All directories named hereafter will be found in this directory, and so the *catkin_ws/src/car_chassis_controller_s22* part will be left out of its location for brevity. The software is handled using the ROS middleware suite, which means you do not need to be in this directory to launch the commands *except* for the

CanBusReader.py script, which has not been integrated into the launch file. Because of this, you must be in the directory of CanBusReader.py to run the script.

To do this, turn the computer on and open a terminal window. Use the change directory command (*cd*) to get into CanBusReader.py's directory:

cd catkin ws/src/car chassis controller s22 /src/pcan

Open another terminal window in this directory by right-clicking on the current terminal window and selecting "New Tab" or "New Window.". You should now have two terminal windows opened in the *pcan* directory. This is done because the CanBusReader.py script is not killable with Control+C, so the terminal it is running in must be closed every time by right-clicking on it and selecting "Close window" manually. To avoid having to change into CanBusReader.py's directory each time you close and reopen the terminal, you can *cd* into the directory in one terminal, then use it to open another in the same directory.

Next, open the Arduino IDE. Open the Brake_Control and Steering_Throttle_Control files, which are located in the *src/arduino* directory. Make sure that each of these files is uploaded to the proper Arduino by uploading them and checking to make sure that the proper Arduino flashes for each upload. The Steering_Throttle_Control file should be uploaded to the Arduino with many DACs on it and the Brake_Control file to the Arduino with a single pin connector for the stepper driver.

The next step is to make sure that the emergency stop button and autonomy switch are disengaged. These are located in the center console of the car. The left E-stop button must be in the up position (twist until it pops up) and the autonomy switch must be towards the right. Both of these switches are wired in series so both need to be in the right position to send signals to the car. Engaging either system cuts power to the relays and returns manual control of the vehicle.

The E-stop button doesn't automatically stop the car; user intervention is necessary. The following is the most important instruction in the user manual. If the E-Stop needs to be flipped during operation, it is imperative to not turn the autonomy back on without first turning the car off. The E-stop and autonomy switch need to be in the correct positions before the car is turned on. Turning autonomy on while the car is on sends a voltage spike to the systems that can burn out the motor in the power steering assembly.

In one terminal, write but do not run the command *roslaunch car_chassis_controller_s22 car.launch*. In the other terminal, write but do not run *python CanBusReader.py*. The launch file will run all necessary programs for teleoperation including operator.py, rosserial for Arduino communication, and joystickControl.py. CanBusReader.py will decode the CAN messages and send them to operator.py.

Run the *car.launch* command. You should see ROS start up, some information text, and then a line saying "*Controller Not Connected: Press A.*" Before pressing A, press both triggers a few times and spam a few buttons like the D-pad and joysticks. This helps to establish the communication between the joystick and the computer and reset the potentiometers in the joystick. After this, press A. The message should change to "*BRAKE NOT ZEROED.*" Now you can run the *CanBusReader.py* command. You should see this terminal window begin to print CAN values, and eventually the other terminal window will say "*Brake Zeroed.*" Once this is printed, the car should be ready for teleoperation. Remember to press the brake before taking the car out of park. The operation of the car uses the left trigger for brake control, right trigger for throttle, and the left stick for steering.

To stop the autonomous driving mode, stop the car using the brakes and put it into park. Kill the *car.launch* terminal with Control+C, and then close the *CanBusReader.py* terminal by

right clicking and selecting "Close." To return the car to non-autonomous mode, press the emergency stop and flip the switch from autonomous mode to non-autonomous mode. Again, it is imperative to turn the car all the way off before switching it back into autonomous mode.

8 CONCLUSIONS AND FUTURE WORK

The team was able to accomplish its main goals and successfully completed a test drive of the system. External electromechanical and wiring systems that allow for the control of the car's steering, brake, and throttle systems have been installed. Software to interface with the external control system, the joystick controller, and the car's internal CAN bus for the teleoperation of the car have been created. PID controllers to allow for feedback control of the steering and braking systems using data from the CAN BUS have been designed. The sensor suite for LiDAR, GPS, IMU, and depth cameras have been mapped and designed. All subsystems are controlled together, meaning the car is now fully drive-by-wire, and is able to drive on the road, albeit slowly and safely in a parking lot.

Future work on the system involves several small fixes to make the system more reliable with more safeguards to prevent damage. Adding the functionality to account for user input in the steering wheel during operation and adding components to the relay circuit to prevent the spike in voltage when the system is turned on are some examples that will help future teams. However, future work on the car will primarily involve making the car fully autonomous. The first step of this process will be in creating more reliable PID controllers for steering, throttle, and braking. Once these subsystems can be more accurately controlled with an external controller, sensors can begin being integrated into the system. This sensor data can be used to

create semi-autonomous processes such as self-parking and speed control by interpreting data from the outside world. Once enough semi-autonomous control loops are integrated into the system future teams can create more autonomous features and eventually a fully autonomous vehicle. It is imperative for future teams to more accurately control the integral driving subsystems and convert the data points recorded by the sensors into a complete image of its surroundings in order to perform logical processes.

Appendix

Appendix A: QFD Chart

	Importance (Customer Needs)	SPECIFICATIONS:	Turning resolution	0 - 60mph acceleration time	Response time for Braking	Response time for throttle	Obstade avoidance distance	Response time for steering	braking distance from 60mph - 0mph	Throttle (speed changing) resolution
		ļ	1	2	3	4	5	6	7	8
Direction of Improvement	\square	ļ	1 - L	4	1	1	0	1	0	+
CUSIOMER NEEDS:		ļ			2	2				-
Dirive-by-wire system	2	ŀ	4	4	5	3	4	4	5	4
Must have steering and overnide mechanical or electrical capabilities	4	ŀ	3	3	2	3	4	3	3	2
bioing dataparancy required of the autonomous system to relay its indugine to passenger.	3	ŀ	-	1	1	,	1	5	-	-
bat can be concerted to or or mapping, but should not be used outside or venice or to improve remete a driving capabilities.	3	ŀ	2	2	2	2		2	3	3
TECHNICAL IMPORTANCE, DIFFICULTY, AND PRIORITY:	-	h	_	-	-	-		-		-
Technical importance (calculated, absolute) Technical importance (calculated, percentage) Technical importance (calculated, rank)			9 10.4% <mark>8</mark>	4 11.2% <mark>4</mark>	L 16.4% 0	33 %0.6	35 %9:6 7	0 13.7% <mark>2</mark>	L 16.4% 0	4% 4% 4%
			0.2							
Units			deg	ø	ø	ø	ŧ	ø	Ŧ	đ
Marginal Target Value				2	0.5	0.5	50	0.5	8	
Ideal Target Value		Ī	_		5.1	1.0	2	17	8	_
Technical Difficulty	Н	- t	2	2	3	3	1	3	4	2
Technical Priority (calculated)			76	82	180	99	35	150	240	98
Technical Priority (calculated, rank)			7	6	2	4	8	3	1	5
			250							



Appendix B: Concept Generation- Functional Decomposition

Sub-Functions			Solution	18	
Actuator	Stepper Motor	Servo Motor	DC Brush Motor	DC Brushless Motor	Hack power-braking system within car
Power	12V Car battery stepped up to	Power Supply	Solar Panel	Portable Power Cell	Wall power
Input control	GUI on onboard computer	Wired Potentiometer	Code positions looping in code	External joystick Controller	
Microcontroller Board	Arduino Mega/Uno	Texas Intruments Microcontoller	FPGA board (XEM8350)	ASM Microcontroller	Parallax P2 Chip
Sensors (feedback control methods)	Sensor suite for measuring speed	Ultrasonic distance sensor mounted behind pedals	Quadrature encoder	Absolute encoder	CAN Bus brake pedal position

Appendix C: Concept Generation- Different Solutions for Braking

Appendix D: Concept Generation- Different Solutions for Throttle

Sub-Functions				Solutions		
Method	Stepper Motor to pull throttle	Servo Motor to pull throttle	Send DigiPot signals to emulate signals from throttle potentiometers	Replace actuator for opening throttle air intake with our own mechanism that we control	Pneumatic actuator to push throttle	
Power	12V Car battery	Power Supply	Solar Panel	Portable Power Cell	Wall power	
Input control	GUI on onboard computer	Wired Potentiometer	Code positions looping in code	External joystick Controller		
Microcontroller Board	Arduino Mega/Uno	Texas Intruments Microcontoller	FPGA board (XEM8350)	ASM Microcontroller	Parallax P2 Chip	
Sensor (Feedback control method)	Triple potentiometers imbedded in throttle pedal	Ultrasonic distance sensor mounted behind pedals	Quadrature encoder	Pitot tube inside throttle assembly to measure air intake	CAN Bus speed reading (spedometer)	Sensor suite (Radar, Lidar, GPS) for measuring speed

Sub-Functions			Solution	15	
Method	Motor built in line with steering column	Send a torque sensor message to the ECU to emulate power steering reading	Exterior motor attached to a belt wrapped around the steering wheel	Replace power steering module with a controllable hydraulic power steering module	
Power	12V Car battery	Power Supply	Solar Panel	Portable Power Cell	Wall power
Input control	GUI on onboard computer	Wired Potentiometer	Code positions looping in code	External joystick Controller	
Microcontroller Board	Arduino Mega/Uno	Texas Intruments Microcontoller	FPGA board (XEM8350)	ASM Microcontroller	Parallax P2 Chip
Sensor (Feedback control method)	Triple potentiometers imbedded in throttle pedal	Sensor suite (Radar, Lidar, GPS) turning radius	Absolute encoder on steering wheel	Accelerometer mounted on car to measure turning	CAN Bus steering wheel position reading

Appendix E: Concept Generation- Different Solutions for Steering

Appendix F: Concept Generation- Other Solutions Info

The **Carnegie Mellon Boss**, a Chevy Tahoe, did have steering, throttle, and brakes controlled with actuators. It was mentioned that normal driving controls are available in case of emergency override, so it is implied that sensors to detect override are present. For internal processing and data computation, the Boss uses a CompactPCI Chassis with 10 Core2Duo Processors.

The **Stanford Stanley**, a Volkswagen Touareg R5 did have steering, throttle, and brakes mechatronics controlled with actuators. Processing was done internally in the car with a system of external computers running Linux software hidden in the trunk. Linux was used due to its excellent networking and time sharing capabilities. Race software was executed on three of the six computers, with the fourth computer to log race data, leaving two idle computers. One of the race computers handled solely video processing and the other two race computers executed all other software polling the sensors and controlling the

brake, steering, and throttle. There was no mention on whether sensors to detect manual override were present (other than an E-stop button), if there was an intuitive graphical interface to show processing, or a physical controller.

MIT's Talos vehicle, a Land Rover LR3, includes servos for throttle and brake actuation. The car has a "Situational Planner" interface, which does show vehicle decision making ahead of time. Processing is done on a Quanta blade server computer system. It was mentioned that there is a "proven and safe method" for switching between manual and autonomous driving, but it is unclear whether it is some sort of E-stop or sensors on the throttle, brake, or steering themselves. There was no mention of a physical controller.

Virginia Tech's Victor Tango car is a Chevy Volt. Very limited information was found about the car but from what was gathered, the steering and throttle/brake systems are not controlled with actuators, but computing was done internally.

Some autonomous vehicles that are currently in the industry include **Volvo's driveMe** system and **NuTonomy's driverless taxi**. Both had steering, brake, and throttle systems controlled by actuators, as well as internal secure processing (nuTonomy used machine learning) but did not have sensors to detect override other than an E-stop.

REFERENCES

Kottayil, N. K. (Ed.). (2021, August 24). *What is an autonomous vehicle? - definition from Techopedia*. Techopedia.com. Retrieved March 21, 2022, from https://www.techopedia.com/definition/30056/autonomous-vehicle

Laukkonen, J. (2020, February 13). *Take a ride in Waymo's self-driving car*. Lifewire. Retrieved March 29, 2022, from https://www.lifewire.com/waymo-self-driving-cars-4171314

PID theory explained. National Instruments. (2020, March 17). Retrieved May 8, 2022, from https://www.ni.com/en-us/innovations/white-papers/06/pid-theory-explained.html

Schwall, M., Daniel, T., Victor, T., Favarò, F., Honhold, H., (2020). Waymo public road safety performance data.

https://storage.googleapis.com/sdc-prod/v1/safety-report/Waymo-Public-Road-Safety-Performan ce-Data.pdf

Shipley, D. (2021, November 17). A Trucker Shortage? Bring on the Robots. Bloomberg.com. Retrieved February 24, 2022, from

https://www.bloomberg.com/opinion/articles/2021-11-17/autonomous-vehicles-could-solve-the-u-s-truck-driver-shortage

Wevolver. (2020). 2020 autonomous vehicle technology report Wevolver. Retrieved from https://www.wevolver.com/article/2020.autonomous.vehicle.technology.report

Jmscslgroup. (2013). Jmscslgroup/catvehicle: A macroscopic multivehicle tesbed and hardware-in-the-loop simulator for autonomous driving. GitHub. Retrieved from https://github.com/jmscslgroup/catvehicle

Undergraduate Thesis Prospectus

How can a 2008 Ford Escape be made into an automated vehicle? (technical research project in Mechanical Engineering)

Unsafe Streets: Are Automated Vehicles the Solution?

(sociotechnical research project)

by

Matthew Deaton

March 3, 2022

technical project collaborators:

Jacob Deane Henry Goodman Vishal Singh Logan Montgomery Alex Pascocello

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Matthew Deaton

Technical advisor:Tomonari Furukawa, Department of Mechanical EngineeringSTS advisor:Peter Norton, Department of Engineering and Society

General Research Problem

How can the danger of driving be reduced in the U.S.? Driving is one of the more dangerous and costly activities in the U.S. (Toups, 2016). Which is a problem, because most of us drive, 227 million (FHA, 2020) out of 332 million (Census Bureau 2021). The invention of things such as airbags, seatbelts, and crumple zones have made collisions more survivable. Regulations that limit speed and that prohibit some distractions have helped too. These laws help prevent accidents from happening. Nevertheless, in 2019 36,096 Americans were killed on U.S. roads in 2019 (NHTSA, 2020). Further safety improvements are therefore necessary.

How can a 2008 Ford Escape be made into an automated vehicle?

How can a 2008 Ford Escape be made into an automated vehicle? This is the general Capstone Technical Research Problem given to my team. I am working on this Problem with Jacob Deane, Henry Goodman, Vishal Singh, Logan Montgomery and Alex Pascocello. My technical advisor is Tomonari Furukawa.

The given Technical Research Problem was determined to be too long to complete in a single academic year. This is because the car has been used for several different types of projects over the years, a lot of work needs to be done before any actual coding of the "self-driving" portion of the project can be worked on. Therefore, my team is to "Modify 2008 Ford Escape to 'Drive by Wire,'" which means that the Ford Escape can be driven by an electronic controller. Without making the Ford Escape able to Drive by Wire, the car cannot be controlled by a computer. Our group is at a minimum to make the care steer by the input of a computer and at maximum be able to shift, accelerate and brake the same way.

The state of the art is significantly more advanced than myself or my team can hope to achieve by modifying our modest 2008 Ford Escape. Companies like Waymo (Waymo n.d.) and Aurora (Aurora n.d.) have been using and testing this technology for years now. As such our Technical Research Problem is more for the sake of mastering autonomous systems than for research or contributing to society.

The only system that we are required to use in our solution to our problem is a software package called "Melodic Melodrama." Melodic Melodrama is a Robotic Operating System (ROS) used to control autonomous systems. We are to link this ROS to the Ford Escape's Common Area Network (CAN) Bus. The CAN Bus is what the Ford Escape uses to communicate between its systems, such as the pedals to the gas and brakes or the steering wheel to the steering actuators.

The current way an operator turns the tires is by turning the steering wheel, which is linked to a sensor that measures the angle of the wheel. This sensor sends information through the CAN to the actuators attached to the tires, which turn them accordingly. Linking the CAN to ROS will allow us to bypass the steering wheel and access the steering actuators directly, thus achieving to ability to drive by wire.

At the end of this semester, if my group succeeds, we will be able to drive the Ford Escape without being in the vehicle at all. The future groups will then mount and program sensors to gather information about the world outside the car. They will then need to program ROS to interpret this information and send it to the CAN using our connection between the two. When this is done the car will be self-driving. Future groups will then refine the code and systems to improve the cars self-driving capabilities.

Unsafe Streets: Are Automated Vehicles the Solution?

How are promoters and critics of automated vehicles competing to influence the future of traffic safety in the U.S.? There were over 38,000 crash deaths in in the U.S. in 2020 (DOT, 2020) *Find statistic on the amount and cost of crashes*. This number is a marked increase of about 6,000 deaths as compared to the approximate 32,000 in 2016 (CDC, 2016 December 5) this is likely due to the advent of Covid-19 (NHTSA, 2021). Participants include developers of self-driving cars (Waymo, n.d.; Aurora n.d.), those worried about potential cyber-attacks (Chowdhury, 2020), skeptics on and opponents of AVs (Roy, n.d.). Automated vehicles have been controversial, especially since an automated Uber killed a pedestrian, Elaine Herzberg, in 2018. (Wired n.d.; Hawkins, 2020).

Waymo was started in 2009 by Google (Waymo n.d.), Aurora was founded in 2017 (Aurora n.d.). The companies use systems of lidar, radar, cameras, programs and computers to create self-driving cars. The technology is still being tested, but Waymo and Aurora's big push for automated technology is that their systems won't get tired, or drunk, or decide to show off in front of the attractive computer program driving next to it. Both companies have spent thousands of hours refining their programs. Waymo is trying to enter the taxi service specifically and has made significant headway. The company logged significantly more hours driven than any other over the last year. It has been taxiing its "trusted testers" around San Francisco and has been doing so since August 2020. The city of San Francisco recently granted Waymo the ability to start charging its riders for rides, provided that there is a human "safety driver behind the wheel while the car is in operation (). Aurora has made partners with two major trucking companies, Paccar and Volvo, and is attempting to break into the trucking industry (). Aurora claims its driver is superior due to its ability of their cars to share information. Stating that the

maps their cars create and share will be extremely effective. However, the companies have yet to release their technology to the public at large.

Waymo and Aurora are two of the most promising AV companies. However, they are not the only companies investing in this technology. Progress is steady, and it is clear the technology is here to stay. The question is how effective it will be. Small pieces of self-driving technology have already been implemented into current vehicles, like lane assist, smart cruise control and aided parallel parking. The technology being created by AV companies has clear benefits, but it is unknown as to whether or not it will fulfill it's intended purpose of being truly independent of human oversight.

Some critics of AVs oppose them on civil rights grounds. Alex Roy, author of the Human Driving Manifesto, claims that AVs threaten to sacrifice personal liberty in the name of safety. He argues that "the very language of self-driving is slavery" (Roy 2018). Roy argues that the advent of AV's will eventually lead to removal of citizen's right to drive. Roy and those who support him claim that the human ability to make decisions is more important than safety. The Human Driving Manifesto is hoping to create a additional Constitutional Amendment, adding a "Right to Drive" alongside the other rights that are guaranteed to U.S. citizens. The Human Driving Manifesto's end goal is to use technology to air human drivers, but not take over driving entirely.

The technology required to truly have self-driving cars is still many years out. It also may not be even possible. Companies like Waymo and Aurora are using computer systems that rely on statistical data gathered from billions of hours spent testing. The problem with this approach is that statistical data while often very accurate, cannot account for every variable in

the same way a human does. Roy and his supporters are acting preemptively hoping to answer the question of whether or not there is a right to drive before it is even asked.

There was a workshop held in 2021 by many IOOs to discuss the role of infrastructure in the event of implementing AVs. Many possible problems were identified by the workshop, anticipated increased load from AV platooning, linking several AV's together and have them drive to the same place, including pavement quality interfering with AVs programs, to name a few. At this stage a majority opinion held by IOOs was that more data is required because there are so many potential problems posed by AVs (Alessandrini 2021). Congress is having difficulty legislating AV implementation. Entire new suites of bills and laws are being written in order to prepare for their possible implementation (Canis 2018).

It is very likely that if AV's are successful and gain popularity that the infrastructure of the US will have to change to accommodate the problems identified in the aforementioned workshop. The cost of a possible complete retooling of all infrastructure may not be worth the convenience of self-driving cares.

AV technology, and the research leading to it is a good thing. It has benefited drivers without being fully realized, and as of right now has not demonstrated itself to be a threat to anyone's liberties. At this current time it is unclear if the technology is actually capable of succeeding, but given then current progress it is not unrealistic to be hopeful.

References

- Chowdhury, A., Karmakar, G., Kamruzzaman, J., Jolfaei, A. and Das, R. "Attacks on Self-Driving Cars and Their Countermeasures: A Survey," in IEEE Access, vol. 8, pp. 207308-207342, 2020, doi: 10.1109/ACCESS.2020.3037705.
- Alessandrini, A., Domenichini, L., & Branzi, V. (2021). The Role of Infrastructure for a Safe Transition to Automated Driving. Amsterdam: Elsevier.
- Aurora. (n.d.). Our Company. Aurora.tech. https://aurora.tech/company
- Canis, B., & Library of Congress (issuing body) (2018). Issues in Autonomous Vehicle Deployment ([Library of Congress public ed.). Washington, D.C.: Congressional Research Service.
- CDC. (2016, July 6). Motor vehicle crash deaths. Centers for Disease Control and Prevention. https://www.cdc.gov/vitalsigns/motor-vehicle-safety/index.html.
- CDC. (2016, December 5). Motor Vehicle Injuries. Centers for Disease Control and Prevention. https://www.cdc.gov/winnablebattles/report/motor.html
- Census Bureau. (2021, October 31). Population Clock. Population Clock. https://www.census.gov/popclock/
- USDOT. (2021, May). Early Estimate of Motor Vehicle Traffic Fatalities in 2020. United States Department of Transportation. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813115
- Federal Highway Administration. (2020, February 28). Highway Statistics 2018. Table DL-22 -Highway Statistics 2018 - Policy | Federal Highway Administration. https://www.fhwa.dot.gov/policyinformation/statistics/2018/dl22.cfm
- Hawkins, A. J. (2020, December 7). Uber's fraught and deadly pursuit of self-driving cars is over. The Verge. https://www.theverge.com/2020/12/7/22158745/uber-selling-autonomous-vehicle-business-aurora-innovation
- Marshall, A. (2020) Uber Gives Up on the Self-Driving Dream. Wired. https://www.wired.com/story/uber-gives-up-self-driving-dream/
- NHTSA. (2021, June 3). 2020 Fatality Data Show Increased Traffic Fatalities During Pandemic. https://www.nhtsa.gov/press-releases/2020-fatality-data-show-increased-traffic-fatalitiesduring-pandemic

Roy, A. (2018, March 5). This Is the Human Driving Manifesto. The Drive. https://www.thedrive.com/article/18952/this-is-the-human-driving-manifesto Toups, D. (2021, January 6). How Many Times Will You Crash Your Car? Forbes. https://www.forbes.com/sites/moneybuilder/2011/07/27/how-many-times-will-you-crashyour-car/?sh=1476e6424e62

Waymo. (n.d.). October 7, 2021 https://waymo.com/company/#story.