

MODELLING HUMAN-AUTOMATION INTERACTIONS IN A HAPTIC  
SHARED CONTROL FRAMEWORK

by

Arjun Vishnu Yeravdekar

A thesis submitted to the faculty of  
The University of North Carolina at Charlotte  
in partial fulfillment of the requirements  
for the degree of Master of Science in  
Mechanical Engineering

Charlotte

2020

Approved by:

---

Dr. Amir H. Ghasemi

---

Dr. Peter T. Tkacik

---

Dr. Jun Xu



## ABSTRACT

ARJUN VISHNU YERAVDEKAR. Modelling Human-Automation interactions in a Haptic Shared Control Framework. (Under the direction of DR. AMIR H. GHASEMI)

This thesis is focused on modeling the interaction between a human driver and an automation system . While numerous companies and academic groups are pushing to develop autonomous vehicles with the aim of freeing up attention for drivers and improving safety on the road; barriers remain for deployment of fully autonomous vehicles, including technical, legal, and social barriers. The challenge of meeting human capabilities for sensing, perceiving, and predicting the environment on the road is formidable, even more challenging is the hand-off problem of how to achieve a smooth transition of control authority between a human driver and automation system. Combining the best capacities of a human driver with the speed, accuracy, and tirelessness of automation will require a shared control framework that is intuitive for the driver.

In this thesis, we explore the interaction of a human driver and an automation system in a haptic shared control framework. In a haptic shared control framework, the human driver and automation system both act on the steering wheel, exerting control on the vehicle but also communicating with each other using haptic cues and signals. Both the human driver and automation system act with limited impedance: the human by biomechanics and the automation system by design, with the use of proportional control.

In this thesis, the interaction between the two agents (i.e., the human driver and automation system) are modeled using a game-theoretic approach. The human and automation system are both modeled with a similar structure. Specifically, the human model consists of a higher-level controller representing his cognitive controller, as well as a lower-level controller representing his biomechanics. Similarly, the au-

tomation system is modeled with a higher-level controller (AI) as well as a lower-level impedance controller. Since the human and automation dynamics can adaptively change by modulating their impedance (lower-level controller), the higher-level controller of the human and automation system is modeled using an adaptive model predictive controller.

When there exist two controllers, that is, the human driver and the automation system, there is the possibility that their objectives in terms of target paths are conflicting, and the corresponding control actions are thereby non-cooperative. To explore the interaction between the driver and the automation system under such conditions, two-games equilibrium strategies known as non-cooperative Nash and Stackelberg are derived, and some simulation results related to these equilibrium types are presented and discussed. The Nash paradigm represents a scheme where both agents act as leaders (i.e., leader-leader) in performing a task. On the other hand, the Stackelberg paradigm presents a case where one agent acts as a leader, while the other acts as a follower. It is shown that for the same impedance, the Stackelberg solution achieves a possibly better path-following performance than the corresponding Nash solution.

## DEDICATION

I would like to dedicate my thesis to my parents.

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Amir H. Ghasemi for his continuous support during my masters study and research, for his encouragement, patience, humility and unrivalled knowledge. I was grateful to have him by my side throughout this two-year journey. Without his inputs and constant support, this work would have been impossible. Dr. Ghasemi not only helped me with the technical guidance, but he also helped me grow as a person.

I would also like to thank my committee members Dr. Peter Tkacik and Dr. Jun Xu for their guidance and precious time. I consider myself lucky for having an opportunity to work under Dr. Tkacik. Under his guidance, I learned the importance of analyzing things with multiple perspective.

I would like to thank my friend and colleague Mr. Vahid Izadi for his inputs and constant support during my research. Lastly, I would like to thank my family and friends who motivated me throughout this masters degree.

## TABLE OF CONTENTS

LIST OF FIGURES	ix
CHAPTER 1: INTRODUCTION	1
1.1. Motivation	2
1.2. Levels of Autonomy	2
1.3. Shared Control	5
1.3.1. Types of Shared Control Framework	6
1.4. Haptic Shared Control	8
1.5. Game Theory	9
1.6. The contribution of the thesis	10
1.7. The outline of the thesis	10
CHAPTER 2: Dynamic Model of a Haptic Shared Control Framework	11
2.1. Modeling the Human Driver	12
2.2. Modeling the Automation System	15
2.3. Human-Automation Steering Model	17
2.4. Vehicle Dynamics	19
2.5. Comprehensive Model of a Haptic Shared Control Framework	22
CHAPTER 3: Design of Impedance Modulation Controller	25
3.1. Model Predictive Controller	25
3.2. Adaptive Model Predictive Impedance Controller	26
3.3. Impedance Modulation Controller Design	27
3.3.1. Modified Non-Negative Least Squares Method	31

CHAPTER 4: Developing Game Theoretic Framework for Intent Negotiation	33
4.1. Non - Cooperative Mode	34
4.1.1. Nash Strategy	37
4.1.2. Stackelberg Strategy	41
CHAPTER 5: RESULTS	47
5.1. Nash Strategy	48
5.1.1. Without Adaptive Law	48
5.1.2. With Adaptive Law	50
5.2. Stackelberg Strategy	52
5.2.1. Without Adaptive Law	53
5.2.2. With Adaptive Law	55
CHAPTER 6: CONCLUSIONS and FUTURE WORK	57
6.1. Conclusion	57
6.2. Future Work	58
REFERENCES	60

## LIST OF FIGURES

FIGURE 1.1: ALFUS Framework	3
FIGURE 1.2: Levels of Autonomy [1]	4
FIGURE 1.3: Shared Control	6
FIGURE 1.4: Types of Shared Control	6
FIGURE 1.5: Input Mixing Shared Control	7
FIGURE 1.6: Risk in Haptic Shared Control	9
FIGURE 2.1: Three torque inputs on the steering wheel with their associated back-drive impedance's	11
FIGURE 2.2: Model of the backdrivable human motor system steering a vehicle. (A) The angular displacement of the driver's hands and the steering wheel are described by the (easily measured) variable $\theta_S$ whereas (B) the driver's steering command $\theta_H$ that acts through the compliance $k_H$ and damping $b_H$ of muscle cannot be measured. (C) An equivalent block diagram with $Z_H = J_H s^2 + b_H s + k_H$ and $Z_S = J_S s^2$ .	14
FIGURE 2.3: Automation Steering Vehicle Model	16
FIGURE 2.4: Human-Automation-Steering Wheel	17
FIGURE 2.5: bicycle model [2]	19
FIGURE 2.6: A General Model of Haptic Shared Control	23
FIGURE 3.1: MNNLS algorithm	32
FIGURE 5.1: Drivers and Automation's target paths	47
FIGURE 5.2: Open loop Nash paradigm without adaptive law: In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for $K_A$ and $B_A$ , when $K_H = K_A$ and $B_H = B_A$	49

FIGURE 5.3: Open loop Nash paradigm without adaptive law: In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 2K_A$  and  $B_H=B_A$  49

FIGURE 5.4: Open loop Nash paradigm without adaptive law: In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $2K_H = K_A$  and  $B_H=B_A$  50

FIGURE 5.5: Open loop Nash paradigm with adaptive law but constant  $K_H$ : In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 1$  and  $B_H = 0.01$  51

FIGURE 5.6: Open loop Nash paradigm with adaptive law and varying  $K_H$ : In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H$  is varying and  $B_H = 0.01$  52

FIGURE 5.7: Open loop Stackelberg paradigm without adaptive law: In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = K_A$  and  $B_H=B_A$  53

FIGURE 5.8: Open loop Stackelberg paradigm without adaptive law: In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 2K_A$  and  $B_H=B_A$  54

FIGURE 5.9: Open loop Stackelberg paradigm without adaptive law: In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $2K_H = K_A$  and  $B_H=B_A$  54

FIGURE 5.10: Open loop Stackelberg paradigm with adaptive law but constant  $K_H$ : In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 1$  and  $B_H = 0.01$  55

FIGURE 5.11: Open loop Stackelberg paradigm with adaptive law and varying  $K_H$ : In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H$  is varying and  $B_H = 0.01$  56

FIGURE 6.1: A real time simulator 58

FIGURE 6.2: Semi-Automated Golf vehicle 59

## CHAPTER 1: INTRODUCTION

People all around the globe are enthusiastic about the development of autonomous vehicles. Autonomous vehicles are controlled by algorithms, which will take much danger out of the equation. These algorithms determine the appropriate stopping distance, distance from another vehicle, and other data that decreases the chance of a vehicle accident. There is no scope for the computer to get distracted, which is a leading cause of road accidents. Studies suggest that 81% of vehicle crashes are because of human errors. Vehicle automation has great potential to improve driver safety and efficiency and reduce driver fatigue.

The conventional view of automated driving is a vehicle that is a hands-off foot-off system where the driver is completely disengaged from the controls and from monitoring the situation. However, until such systems are available, the driver will need to be in the loop, ready and capable of safely taking over control at all times. A shared control system, where an automated system and a driver concurrently control the vehicle, might accomplish this requirement.

In haptic shared control, the human and automation are physically coupled, and this coupling allows them to exchange the control authority dynamically. The driver is not only aware of the actions of the system but can also choose to overrule the system's activity. In this research, we aim to develop a model for cooperative and non-cooperative interactions between the human driver and the automation system using game theory.

## 1.1 Motivation

Semi-autonomous vehicles can keep in lane and even change lanes, and they may also be able to park themselves, but they are not driver-less/Autonomous. An autonomous vehicle is a vehicle that can drive itself without any human conduction. In most cases, drivers must keep their hands on the wheel at all times. These types of vehicles hold great potentials for both military and commercial applications. From the military perspective, they increase personnel safety, increase mission performance, and reduce cost. Similarly, from a commercial point of view, autonomous vehicles increase road safety, life quality, and increase mobility.

Although it is not clear by using autonomous vehicles to what extent human lives can be saved, it's obvious that human-driven vehicles come at a very high cost in terms of danger. According to the U.S. Department of transportation, the use of autonomous vehicles can save a significant amount in many different aspects like health insurance and vehicle insurance associated with accident recovery alone. The use of Autonomous vehicles will also help to reduce traffic congestion, which in return will reduce commute time for drivers in the high traffic area, thereby reducing gasoline consumption. Disabled individuals can now travel with more freedom and enhanced mobility. They will not have to rely on public transport and assistance from others to get around.

## 1.2 Levels of Autonomy

In recent years, the use of unmanned systems has increased a lot ranging from battlefields to the mars operation. Due to this increase in the number of programs for developing unmanned systems, there was a growing need for characterizing the autonomy of these systems. This gave birth to 'Autonomy Level for Unmanned Systems' (ALFUS). ALFUS provides a complex and comprehensive measure of autonomy level and autonomous performance for a specific mission and environment. It focuses

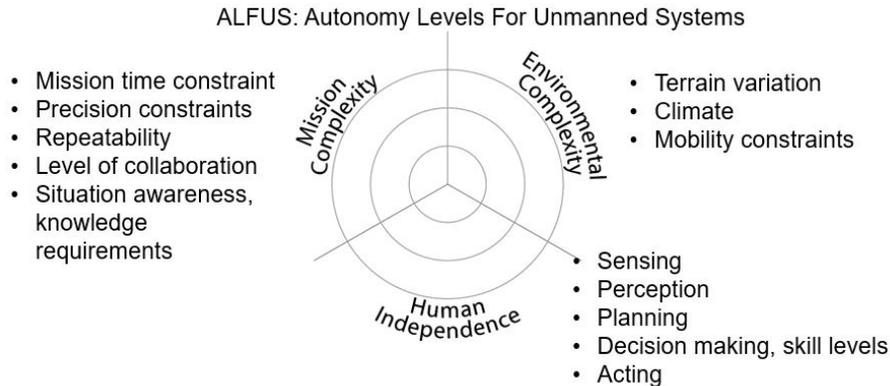


Figure 1.1: ALFUS Framework

on an unmanned system's ability of sensing, perceiving, analyzing, communicating, planning, decision-making, and acting to achieve its goals as assigned by its human operator.

As shown in figure 1.1 ALFUS framework defined three different levels for any unmanned systems: These levels are mission complexity, environment complexity, and human independence. For instance, an autonomous vehicle should be able to address missions with a time constraint, consider the terrain variation and mobility constraints along with proper planning with a particular skill set to complete the mission.

ALFUS was initially presented at the 2004 International Society for Optics and Photonics (SPIE) Defence and Security Symposium since then it has been continuously developed and refined. ALFUS uses the three-axis method of the Contextual Autonomous Capability for accurately assessing the autonomy level. Each axis refers to a metric group, which is mission complexity, environmental complexity, and human independence. For a given mission and environment, metrics are measured for the mission complexity, environmental complexity, and human independence, and these metrics are combined to form a level of autonomy. It consists of a 0 to 10 numeric scale to decide the autonomy level for a given autonomous vehicle, where 0 is the fully human-driven vehicle and 10 being a fully autonomous vehicle [3]. But ALFUS

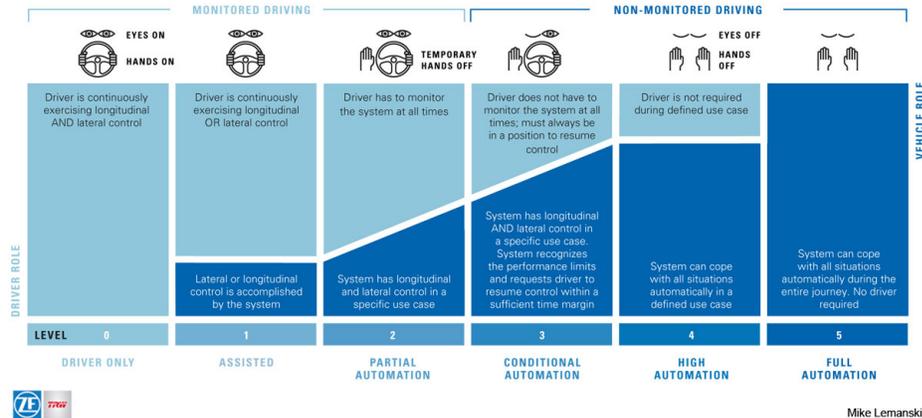


Figure 1.2: Levels of Autonomy [1]

framework has some drawbacks. It allows the metrics to be standardized in scoring scales, which causes subjective evaluation and criteria to influence the results across different robots. It integrates the metrics for a concise set of indices for the autonomy level, and it was mainly developed army's unmanned ground vehicles.

Recently the Society of Automotive Engineers proposed a five-level classification for autonomous vehicles [1]. Not all the vehicles running on roads are alike; there can be few manually driven vehicles, while few can be semi-autonomous, and few can be autonomous. Thus there are different levels of autonomy. To make it easy for the government, the users and for the automotive engineers to get a better understanding of the control authority and gain a better knowledge of this new technology, SAE defined five different levels of automobile autonomy which is as given below: [1]

- 1] Level 0: The vehicle is manually driven with no robot intervention.
- 2] Level 1: A single function is automated, but has nothing to do with the driving environment, for example, cruise control.
- 3] Level 2: Acceleration, Deceleration, and steering are automated, uses sensory input from the environment to make decisions, but still, the driver is responsible for the safe operations of the vehicle. For example, lane tracking and lane changing systems, and also collision-avoidance braking.
- 4] Level 3: In this level, all the safety functions are automated, but the driver is

still required to kick and take control in case of an emergency. This is the most controversial level because it requires the human driver to be present continuously in the loop even though the vehicle is doing everything. Many automakers are thinking of skipping the level 3 and going directly to level 4 because if the human driver loses his/her attention from the task at hand, then it might lead to catastrophic results. For example, the autopilot mode in the Tesla vehicle.

5] Level 4 and Level 5: These levels handle entirely autonomous vehicles. The robot is responsible for making all the decisions. The only difference between 4 and 5 is that level 4 is restricted to city, suburban, and highway driving while the level 5 vehicles can handle every type of situation.

### 1.3 Shared Control

Vehicle automation has the potential to improve driving safety and efficiency and reduce driver fatigue. The conventional view of automated driving is a vehicle that is hands-off feet off system where the driver is completely disengaged from the controls and from monitoring the situation. However, until such systems are available, the driver will need to be in the loop, ready and capable of safely taking over control at all times.

From figure 1.3 we can see that autonomous vehicle covers a short range of mission complexity and environmental complexity but is human independent. On the other hand, the manually driven vehicle because of the experience, decision-making capabilities, judgment is capable of handling more complex scenarios very effectively but is human dependent. We want a system that can address a wide range of mission complexity and environmental complexity and should also be human independent. The question is then until we have this perfect autonomous vehicle, how can we combine the best capabilities of the human with the best of the automation? Studies have shown that Shared Control Frameworks have a high potential to solve this problem [4–9].

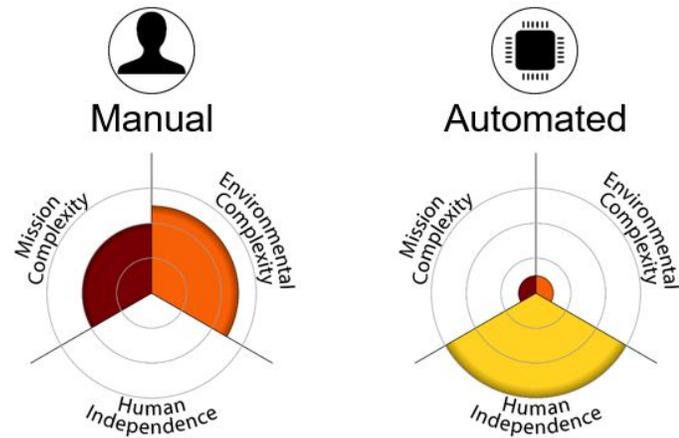
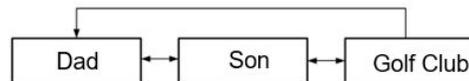


Figure 1.3: Shared Control

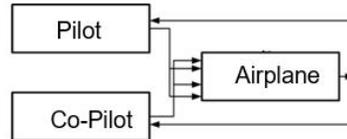
- Supervisory Control



“Hit the ball there”



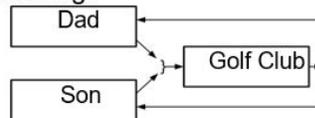
- Input Switching:



“You/I have control”



- Continuous Input Switching:



“I help you swing”



Figure 1.4: Types of Shared Control

### 1.3.1 Types of Shared Control Framework

A wide range of shared control frameworks has been proposed. One of the new shared control frameworks is the supervisory control framework, where the human is acting like a boss and responsible for decisions, and the automation is responsible for performing the task. As we can see from figure 1.4, dad acts as a leader while the son is responsible for performing the task. The golf club is the vehicle in our case.

Another type of shared control framework is switching control. Active safety is an example of such a framework. Human is primarily responsible for performing the task.

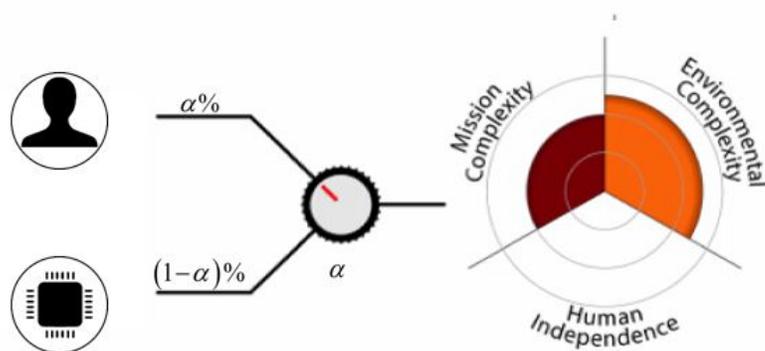


Figure 1.5: Input Mixing Shared Control

However, in a dangerous situation, the automation kicks in and takes the control, for example, the Traction control system, which gets activated when the throttle input and engine torque are mismatched to road surface conditions. Another example of switching control is Autopilot. Where the automation is primarily responsible for the task and in dangerous situations, humans can kick in to take control, as shown in figure 1.4.

But there is one major problem with switching authority strategy that it cannot deal with automation failure when the automation is unaware that it is failed, thus to increase the human's awareness, frameworks have been proposed in which both the agents have some level of control authority, and it is continuously transitioning between the human and automation and have been implemented in two ways: Input-mixing shared control and the haptic shared control. As we can see in figure 1.4 Dad and son both have some control of the Golf club, and therefore, they can edit the other one's movement if necessary.

Input-Mixing shared control, as shown in figure 1.5, the vehicle is controlled based on some weighted combination of the automated controller input and driver input. If this alpha is controlled by the human driver, then the human driver cannot relax, he will always be there in the loop. So generally, this alpha is governed by the automation

system. But if the automation system fails or it sets a wrong value of  $\alpha$ , then the human driver cannot intervene and change the value of  $\alpha$ ; this might lead to a fatal accident.

An example is a lane-keeping assistance in which the desired tire angle is controlled by a steer-by-wire system (steer-by-wire system replaces the mechanical linkage between the column and rack with actuators which supports autonomous and manual driving), which combines the drivers desired steering angle with the steering angle from the assistance system. In other words, when the drivers' actions agree with the goal of the assistance system, the system generates no additional steering input. But when the driver disagrees with the assistance system (i.e., steers out of the lane), additional steering input is generated by the steer-by-wire system so that the command to the tires will ensure good lane-keeping performance. One major drawback of this system is that the driver cannot overrule the system, i.e.,  $\alpha$  is modulated by the automation. So if the automation fails, then the driver cannot gain the authority.

#### 1.4 Haptic Shared Control

In haptic shared control, the human and automation are physically coupled, and this coupling allows them to exchange the control authority dynamically. The driver is not only aware of the actions of the system but can also choose to overrule the activity of the system. In this framework, the drivers' hands remain on the steering wheel while the automation system exerts control through a motor on the steering column. By haptic feedback or feel of touch, the driver can monitor the automation's actions and the automation's authority (impedance) thus the higher-level controller (brain) can afford to relax while the lower level controller (human body) can still be in the loop sensing the output of the automation. When the driver wants to take over, he can impose torques and can express his desire for increased authority by increasing his impedance. The automation system can be designed to monitor the drivers' actions and impedance and either yield authority or retain authority as a

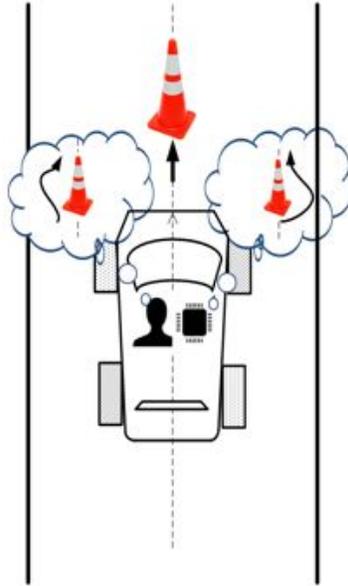


Figure 1.6: Risk in Haptic Shared Control

function of sensed threats to safety.

One of the significant risks associated, as shown in figure 1.6 with haptic shared control framework, is when both human and automation are detecting an obstacle but decides a different path for avoiding it, that is they have different intents for a fixed impedance value. This situation will result in the vehicle going exactly straight into the obstacle, causing an accident. We have developed models for interactions between the human driver and the automation system for such situations using game theory.

## 1.5 Game Theory

In dynamic games, players can condition their optimal actions on what other players have done in the past. Players can observe the actions of other players before deciding upon their optimal response. A dynamic game can be classified based on the mode of the play: whether the two agents are non-cooperating and only thinking about their primary goal or its cooperative. The player's strategy: An example of the player's strategy is Nash strategy where both the agents (driver and automation) wants to

be a leader, or we can say that both the players give their output at the same time, while in a Stackelberg strategy one agent (human) acts as an initiator while the other agent acts as the conductor. By conductor agent, we mean an agent who considers the optimal input of the initiator to determine its control input. In Stackelberg's strategy, the conductor waits for the leader to give his/her optimal output and, after a short delay, gives his optimal output. The players are defined as possessing the open-loop information pattern when only the initial states of the game are known to them. If the dynamic states or outputs are available during the gameplay, players are considered as having the closed-loop information pattern.

## 1.6 The contribution of the thesis

The contribution of this thesis is

- (1) Developed a comprehensive mathematical model for a haptic shared control framework considering the back-drivability of the human and automation system.
- (2) Analyzed the differences in interaction between a driver and an automation system for different shared information patterns.

## 1.7 The outline of the thesis

Modeling of a comprehensive system that includes the human driver, automation driver, steering wheel, and the vehicle is mentioned in Chapter 2, The control theory and the types of control logic used to control the comprehensive model of a haptic shared control framework are described in Chapter 3. Once we know about the controller and the plant, we will go forward with modeling the interactions between the two agents using the control theory and game theory, which is mentioned in Chapter 4. Numerical results are presented in the Results Chapter 5, Which is followed by the Conclusion and Future Works.

## CHAPTER 2: Dynamic Model of a Haptic Shared Control Framework

Vehicles generally operate in closed-loop with the driver. Thus, realistic driver steering control models are needed to allow dynamic vehicle behavior to be optimized. Macadam [10,11] used predictive control theory to derive an optimal preview steering controller. A linear vehicle model was assumed, and a time-invariant controller was derived, which minimized a cost function involving lateral path error. The preview distance could be specified, and the controller was optimized on the basis that the steering angle was held constant over the preview distance. Peng [12] extended Macadam's use of predictive control theory to include path-following yaw displacement errors in the cost function and to allow for non-constant steer angle control. At present, there is little understanding of how the human neuromuscular system operates in closed-loop tracking tasks. The best that is likely to be achieved in the near future is the identification of model structures that closely match measured human responses. Understanding the neural processes by which the human achieves the control is likely to remain a challenge for some time.

Figure 2.1 highlights in a high-level schematic how three entities each impose a torque on the steering column: a human driver through his hands, and automation system through a motor, and the road through the steering linkage. But each of these entities imposes a torque through a certain back-drive impedance, insofar that they are not ideal torque sources. The human is certainly back driveable, and his back-drive impedance we label  $Z_H$ . To indicate

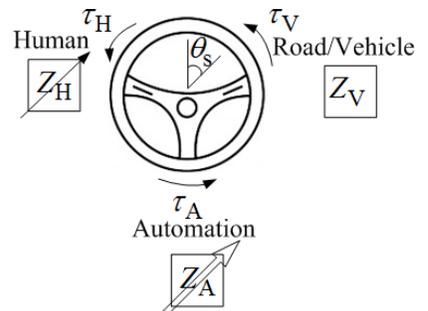


Figure 2.1: Three torque inputs on the steering wheel with their associated back-drive impedance's

that driver impedance varies with changes in grip on the steering wheel, use of one hand or two, muscle co-contraction, or posture changes, we have drawn an arrow through  $Z_H$ . The road and vehicle steering linkage make up a load for the human (and/or automation) that may be described using the impedance  $Z_V$ . By design (possibly involving power steering assist),  $Z_V$  is approximately matched to  $Z_H$ . The crux of this thesis lies in the design of a back-drive impedance  $Z_A$  for the automation that is modest rather than infinite. That is, the automation is not designed to behave as an ideal torque source. Rather, the automation imposes its command torque  $\tau_A$  through an impedance  $Z_A$  that is approximately matched to the human impedance  $Z_H$  (and for that matter matched to  $Z_V$ ). But we further propose that  $Z_A$  shall be varied under control of the automation so as to express the automation's current level of control authority.

In below, a detailed model of a haptic shared control paradigm is derived. Specifically, we derive the equations of motions for four main elements of the haptic shared control paradigm: human driver, automation system, steering wheel (interface), and the vehicle.

## 2.1 Modeling the Human Driver

To describe how a driver places his or her hands on the steering wheel and sets the angle of the tires, the altogether simplest driver model would consist of a motion source acting on the steering wheel. However, we have to ask whether excursions would be produced in the hands and steering wheel by torques arising from tire/road interaction or other sources. Certainly, when the tires encounter an unexpected curb or rut in the road, excursions from the nominal steering angle can be observed. That is, the driver's hands and arms are back driven by the torques arriving at the steering wheel. Thus a motion source model for the driver must be rejected. Even the self-centering torque will produce excursions if it is unexpectedly removed. Also, while power steering may reduce the torque response of the vehicle, that torque is by design

not eliminated in steering assist system designs as it vehicles valuable information for the driver. With these observations in mind, we adopt a finite impedance model for the driver.

The time-varying impedance (or admittance) that describes the response of a driver’s hands and arms to perturbation at the steering wheel owes its origins to the passive dynamics, reflex responses, and volitional responses of the driver’s biomechanics and Central Nervous System (CNS). In particular, the passive response is due to the inertia in the hands and arms and the elastic and damping properties in muscle and other tissues. Note that the passive dynamics is a strong function of posture and the state of muscle contraction or co-contraction [2, 13].

Consider a steering wheel with inertia  $J_S$  and steering angle  $\theta_S$  as shown in Fig. 2.2(A). A driver imposes a torque  $\tau_H$  on the steering wheel through his/her hands while the vehicle/tire/road imposes a torque  $\tau_V$  through the steering column. To describe the finite impedance of the driver, we use a simple second-order model comprising inertia  $J_H$ , stiffness  $k_H$ , and damping  $b_H$ , taking our lead from previous characterizations of human backdrive impedance [14–16]. Values for these elements can be determined by system identification and describe the backdrivability or “give” of the driver’s hands under load. In the interest of simplicity, we assume the contributions of reflex loops are included in the backdrive impedance described by elements  $J_H$ ,  $k_H$ , and  $b_H$ .

To describe the steering command issued by the driver’s CNS, we equip the spring-mass-damper model with a proximal motion source  $\theta_H(t)$  as shown in Fig. 2.2(B). Muscles with elasticity  $k_H$  and damping  $b_H$  are driven under neural control to manipulate, act, and perform mechanical work. The motion source  $\theta_H(t)$  can be considered a kind of desired steering angle or virtual trajectory, as the spring  $k_H$  has zero rest length and the hand/wheel position  $\theta_S$  will equal  $\theta_H$  under no-load conditions ( $\tau_V = 0$ ) and after transients have died out. This model allows us to describe the role of the

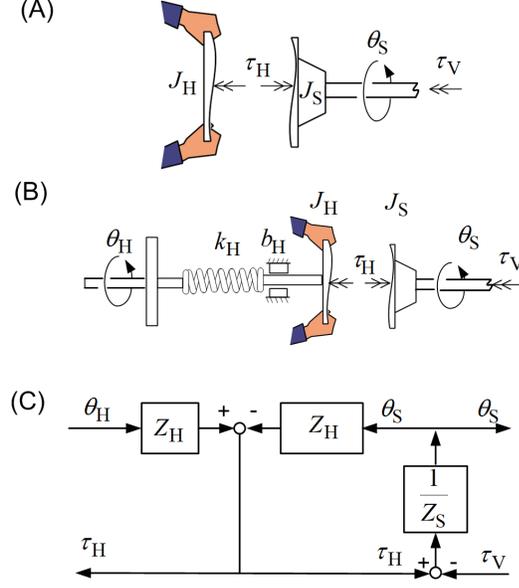


Figure 2.2: Model of the backdrivable human motor system steering a vehicle. (A) The angular displacement of the driver's hands and the steering wheel are described by the (easily measured) variable  $\theta_S$  whereas (B) the driver's steering command  $\theta_H$  that acts through the compliance  $k_H$  and damping  $b_H$  of muscle cannot be measured. (C) An equivalent block diagram with  $Z_H = J_H s^2 + b_H s + k_H$  and  $Z_S = J_S s^2$ .

driver's body in the loop, both as a dynamic subsystem that becomes coupled to the vehicle dynamics, and as a dynamic subsystem, whose sensing capacity relies in part on its states being excited by power transmitted from the steering wheel.

The equations of motion for the steering system are then

$$\begin{aligned}
 J_S \ddot{\theta}_S &= \tau_H - \tau_V = -J_H \ddot{\theta}_S - b_H (\dot{\theta}_H - \dot{\theta}_S) + k_H (\theta_H - \theta_S) - \tau_V \\
 \Rightarrow Z_S \theta_S &= Z_H (\theta_H - \theta_S) - \tau_V
 \end{aligned} \tag{2.1}$$

where  $Z_S \triangleq (J_S + J_H) s^2$  describes the steering wheel inertia plus the human's inertia and  $Z_H \triangleq b_H s + k_H$  represents the effective impedance of the hand and arms on the steering wheel.

Figure 2.2(C) shows a block diagram that includes the dynamics of the human model and the steering wheel. It follows from Eq. (2.1) that the steering angle  $\theta_S$  is a function of  $\theta_H$  (and  $Z_H$ ) from the driver and  $\tau_V$  from the vehicle and environment.

The transfer function from the two inputs  $\theta_H$  and  $\tau_V$  to the response  $\theta_S$  is

$$\theta_S = \frac{Z_H}{Z_H + Z_S} \theta_H - \frac{1}{Z_H + Z_S} \tau_V \quad (2.2)$$

For this research we have assumed that there are no disturbances in the system from the vehicle and the environment. Thus the above equation can be rewritten as;

$$\theta_S = \frac{Z_H}{Z_H + Z_S} \theta_H \quad (2.3)$$

In a series of experiments, Pick and Cole characterized the backdrive impedance  $Z_H$  of a driver's hands on the steering wheel [13]. These characterizations included electromyographic (EMG) measurements and revealed a significant range of impedance that varies directly with muscle co-contraction. A driver can also vary impedance by modulating reflex gains and changing posture [2, 13]. To develop adaptive haptic support, which considers individual and changing driving behavior, it is necessary to identify the time-varying human biomechanics in real-time. Several approaches, including wavelet transforms, recursive least squares, and linear time-varying parameter estimation techniques, have been proposed to estimate human backdrive impedance [17–19]. Nominally, identification of the driver's impedance would require excitation signals (pulses or sinusoids). However, such signals might be annoying to the driver or, worse, perturb steering control. An alternative approach is to determine a correlation between the impedance  $Z_H$  and the driver's grip force and use this correlation to derive an on-line estimate of human biomechanics [16, 20].

## 2.2 Modeling the Automation System

Next, the equations for the automation system are derived. We consider the automation system to have the same structure as the human driver. A high-level controller that generates the desired steering angle  $\theta_A$  and the low-level controller that

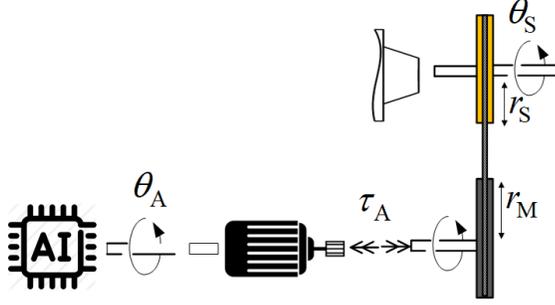


Figure 2.3: Automation Steering Vehicle Model

derives the motorized steering wheel. Considering no human driver, the equations of motion for the motorized steering wheel is

$$J_A \ddot{\theta}_M = \tau_A - F_1 r_M, \quad (2.4)$$

where  $J_S$  and  $J_M$  is the steering wheel and motor inertia, respectively;  $\theta_S$  and  $\theta_M$  are the steering angle and motor angle of rotation and  $r_S \theta_S = r_M \theta_M$  is the kinematic constraint. We define  $Z_S \triangleq J_S s^2$ .

Next, let  $\tau_A$  be

$$\tau_A = \tilde{k}_A (\theta_A - \theta_S) - \tilde{b}_A (\dot{\theta}_S), \quad (2.5)$$

where  $\tilde{k}_A$  and  $\tilde{b}_A$  are proportional and derivative gains. We define  $Z_A \triangleq \left(\frac{r_S}{r_M}\right)^2 J_M s^2 + b_A s + k_A$  and  $k_A \triangleq \frac{r_S}{r_M} \tilde{k}_A$  and  $b_A \triangleq \frac{r_S}{r_M} \tilde{b}_A$ .

The transfer function from the two inputs  $\theta_H$  and  $\theta_A$  to  $\theta_S$  is

$$\theta_S = \frac{Z_A}{J_S s^2 + Z_A} \theta_A. \quad (2.6)$$

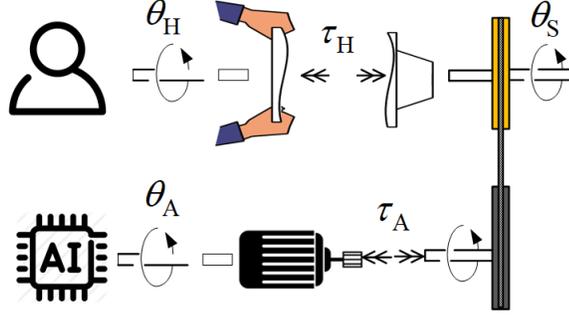


Figure 2.4: Human-Automation-Steering Wheel

### 2.3 Human-Automation Steering Model

Next, we aim to combine the models developed from the human model and automation model to derive the equations of motions for the human-automation-steering wheel system. Note that for the rest of this thesis, we assume  $J_H$  and  $J_A$  is negligible with respect to  $J_S$ .

The equations of motion for the steering wheel when two agents are acting simultaneously can be written as;

$$J_S \ddot{\theta}_S + b_S \dot{\theta}_S = \tau_A + \tau_H \quad (2.7)$$

where  $\tau_H$  and  $\tau_A$  are the human and automation agents torque applied on the steering wheel respectively. But we know that  $\tau_H = Z_H(\theta_H - \theta_S)$  and  $\tau_A = Z_A(\theta_A - \theta_S)$ , thus we can write;

$$J_S \ddot{\theta}_S + b_S \dot{\theta}_S = Z_A(\theta_A - \theta_S) + Z_H(\theta_H - \theta_S) \quad (2.8)$$

Substituting the value for human impedance  $Z_H = b_H s + k_H$  and  $Z_A = b_A s + k_A$  we

get;

$$J_S \ddot{\theta}_S + b_S \dot{\theta}_S = b_S(\dot{\theta}_H - \dot{\theta}_S) + k_H(\theta_H - \theta_S) + b_A(\dot{\theta}_A - \dot{\theta}_S) + k_A(\theta_A - \theta_S) \quad (2.9)$$

We can represent the above equation using a state-space representation. State-space representation in control theory is used to represent any mathematical model.

$$\begin{aligned} \begin{Bmatrix} \dot{\theta}_S(t) \\ \ddot{\theta}_S(t) \end{Bmatrix} &= \begin{bmatrix} 0 & 1 \\ \frac{-k_H - k_A}{J_S} & \frac{-b_H - b_A - b_S}{J_S} \end{bmatrix} \begin{Bmatrix} \theta_S(t) \\ \dot{\theta}_S(t) \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_H}{J_S} & \frac{b_H}{J_S} & \frac{k_A}{J_S} & \frac{b_A}{J_S} \end{bmatrix} \begin{Bmatrix} \theta_H(t) \\ \dot{\theta}_H(t) \\ \theta_A(t) \\ \dot{\theta}_A(t) \end{Bmatrix} \\ Y_S(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{Bmatrix} \theta_S(t) \\ \dot{\theta}_S(t) \end{Bmatrix} \end{aligned} \quad (2.10)$$

This state-space representation can be simplified to

$$\begin{aligned} \dot{x}_S(t) &= A_S x_S(t) + B_S U_S(t) \\ Y_S(t) &= C_S x_S(t) \end{aligned} \quad (2.11)$$

where

$$A_S = \begin{bmatrix} 0 & 1 \\ \frac{-k_H - k_A}{J_S} & \frac{-b_H - b_A - b_S}{J_S} \end{bmatrix} \quad B_S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_H}{J_S} & \frac{b_H}{J_S} & \frac{k_A}{J_S} & \frac{b_A}{J_S} \end{bmatrix} \quad C_S = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad U_S = \begin{Bmatrix} \theta_H(t) \\ \dot{\theta}_H(t) \\ \theta_A(t) \\ \dot{\theta}_A(t) \end{Bmatrix}$$

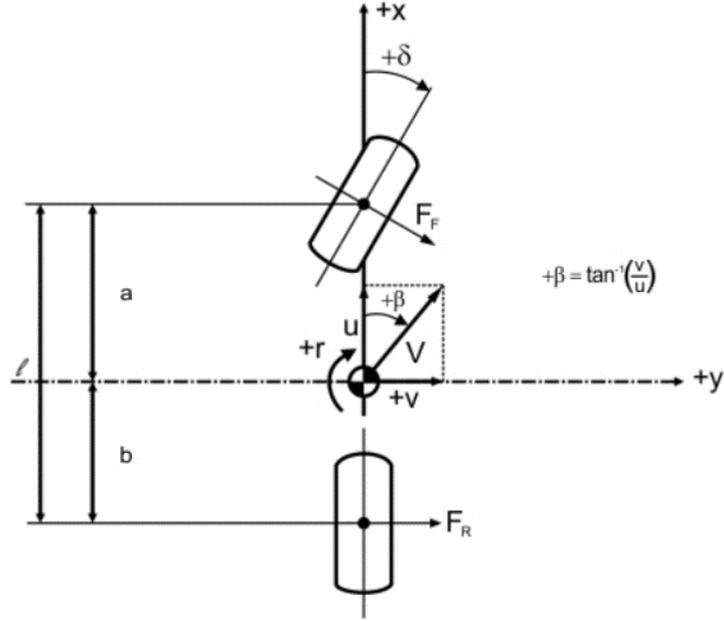


Figure 2.5: bicycle model [2]

## 2.4 Vehicle Dynamics

The directional vehicle dynamics are represented using the well-known yaw-side slip bicycle model shown in 2.5. Yaw-side slip bicycle model is based on Newton's second law of motion and moment balance about the  $z$  axis.

$$I\ddot{\varphi} = a \times F_{y_f} - b \times F_{y_r} \quad (2.12)$$

$$m \times a_y = F_{y_f} + F_{y_r} \quad (2.13)$$

where  $F_{y_f}$  and  $F_{y_r}$  are the lateral tire forces of the front and rear,  $a$  and  $b$  are the distances of the front and rear tire respectively from the C.G.,  $a_y$  is the inertial acceleration at C.G. in  $y$  direction. There are two terms which contribute to  $a_y$ , they are the  $\ddot{y}$  in the  $y$  direction itself and the centripetal acceleration which is  $V \times \dot{\varphi}$  (centripetal acceleration is rate of change of tangential velocity  $= \frac{V_x^2}{r} = \frac{V_x \times V_x}{r} = V_x \times \omega = V_x \times \dot{\varphi}$  where  $V_x$  is the longitudinal velocity and  $\dot{\varphi}$  is the heading angle). Lateral force is di-

rectly proportional to the slip angle( provided that slip angles are small). Slip angle is the difference between the orientation of tire and orientation of velocity vector.

Thus,  $F_{y_f} = C_f \times (\theta_s - \theta_{vf})$  and  $F_{y_r} = C_r \times -\theta_{vr}$ , where  $C_f$  and  $C_r$  are the cornering stiffness of the front axle and rear axle respectively.

$$\tan \theta_{vf} = \frac{V_y + l_f \times \dot{\omega}}{V_x} \quad (2.14)$$

$$\tan \theta_{vr} = \frac{V_y - l_r \times \dot{\omega}}{V_x} \quad (2.15)$$

Considering small slip angles  $\theta_{vf}$  and  $\theta_{vr}$ , and substituting these values of  $F_{y_f}$  and  $F_{y_r}$  in the (2.12) and (2.13) We will get the equation of motion in the first order matrix form with 2 states variable  $\dot{v}$  and  $\dot{\omega}$ . Yaw displacement is the direction of the longitudinal axis of the vehicle and its rate of change is just the yaw velocity  $\dot{\varphi} = \omega$ . So now the yaw angle is also evaluated with the lateral and yaw velocities. In the synthesis of the steering controllers it is usually necessary to have expressions for the lateral and yaw deviations of the vehicle from the road path so that the path following error can be incorporated into a cost function. The yaw deviation presents no problem however the lateral deviation is more difficult since it depends on X and Y and the trigonometric terms which makes a linear calculation more difficult. So, we assume that yaw displacements are small which eliminates the trigonometric functions and further assume that  $V_y \ll V_x$ . To allow the nominal direction of the road path to be other than the X direction we define second set of x-y ground fixed axis. The x axis is aligned with the nominal direction of the road path so that the vehicle has small angular displacements from the x axis. We get,  $y = V_x \times \varphi + V_y$  So now we can also evaluate lateral displacement along with yaw angle, lateral and yaw velocities. This equation can be also written in matrix form as follows:

$$\begin{Bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \\ \dot{y}(t) \\ \dot{\psi}(t) \end{Bmatrix} = \begin{bmatrix} \frac{-(C_f+C_r)}{Um} & \frac{-(l_a C_f - l_b C_r)}{Um} - U & 0 & 0 \\ \frac{-(l_a C_f - l_b C_r)}{UI} & \frac{-(l_a^2 C_f + l_b^2 C_r)}{Um} & 0 & 0 \\ 1 & 0 & 0 & U \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \\ y(t) \\ \psi(t) \end{bmatrix} + \begin{Bmatrix} \frac{C_f}{mG} \\ \frac{l_a C_f}{IG} \\ 0 \\ 0 \end{Bmatrix} \theta_S(t) \quad (2.16)$$

where  $v(t)$  is the lateral velocity of the vehicle in the direction of vehicles lateral axes,  $\omega(t)$  is the yaw velocity of the vehicle with respect to the ground,  $y(t)$  is the lateral displacement of the vehicle in-ground fixed axes, assuming small  $\psi(t)$ ;  $\psi(t)$  the yaw angle of a vehicle with respect to ground;  $\delta(t)$  is the handwheel angle;  $G$  is the steering gear ratio (handwheel angle/road wheel angle);  $U$  is the velocity of the vehicle in the direction of vehicles longitudinal axis (as measured by an observer in the vehicle);  $m$  is the mass;  $I$  is the polar inertia;  $l_a$  is the distance from the center of mass to the front axle;  $l_b$  is the distance from the center of mass to rear axle;  $C_f$  is the front tire cornering stiffness; and  $C_r$  is the rear tire cornering stiffness.

The lateral displacement and the yaw angle can be defined as the outputs using;

$$Y_v(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} v(t) \\ \omega(t) \\ y(t) \\ \psi(t) \end{Bmatrix} \quad (2.17)$$

In matrix notation (2.5) is expressed as;

$$\begin{aligned} \dot{x}_v(t) &= A_v x_v(t) + B_v \theta_S(t) \\ Y_v(t) &= C_v x_v(t) \end{aligned} \quad (2.18)$$

where

$$A_v = \begin{bmatrix} \frac{-(C_f+C_r)}{U_m} & \frac{-(l_a C_f - l_b C_r)}{U_m} - U & 0 & 0 \\ \frac{-(l_a C_f - l_b C_r)}{UI} & \frac{-(l_a^2 C_f + l_b^2 C_r)}{U_m} & 0 & 0 \\ 1 & 0 & 0 & U \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad B_v = \begin{bmatrix} \frac{C_f}{mG} \\ \frac{l_a C_f}{IG} \\ 0 \\ 0 \end{bmatrix}$$

$$C_v = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad x_s = \begin{bmatrix} v(t) & \omega(t) & y(t) & \psi(t) \end{bmatrix}^T$$

## 2.5 Comprehensive Model of a Haptic Shared Control Framework

So till now, we have looked at the steering model and the vehicle model, so now we will have to combine these two system models into a big system model, which comprises both the plant models, as shown in figure 2.6. This system will have the human intention  $\theta_H$  and automation intention  $\theta_A$  as the inputs while its output will be lateral displacement  $y$  and yaw angle  $\psi$ . The state-space model can be represented as follows;

$$\begin{pmatrix} \dot{v}(t) \\ \dot{\omega}(t) \\ \dot{y}(t) \\ \dot{\psi}(t) \\ \dot{\theta}_S(t) \\ \ddot{\theta}_S(t) \end{pmatrix} = \begin{bmatrix} \frac{-(C_f+C_r)}{U_m} & \frac{-(l_a C_f - l_b C_r)}{U_m} - U & 0 & 0 & \frac{C_f}{mG} & 0 \\ \frac{-(l_a C_f - l_b C_r)}{UI} & \frac{-(l_a^2 C_f + l_b^2 C_r)}{U_m} & 0 & 0 & \frac{l_a C_f}{IG} & 0 \\ 1 & 0 & 0 & U & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{-k_H - k_A}{J_S} & \frac{-b_H - b_A - b_S}{J_S} \end{bmatrix} \begin{pmatrix} v(t) \\ \omega(t) \\ y(t) \\ \psi(t) \\ \theta_S(t) \\ \dot{\theta}_S(t) \end{pmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_H}{J_S} & \frac{b_H}{J_S} & \frac{k_A}{J_S} & \frac{b_A}{J_S} \end{bmatrix} \begin{pmatrix} \theta_H(t) \\ \dot{\theta}_H(t) \\ \theta_A(t) \\ \dot{\theta}_A(t) \end{pmatrix} \quad (2.19)$$

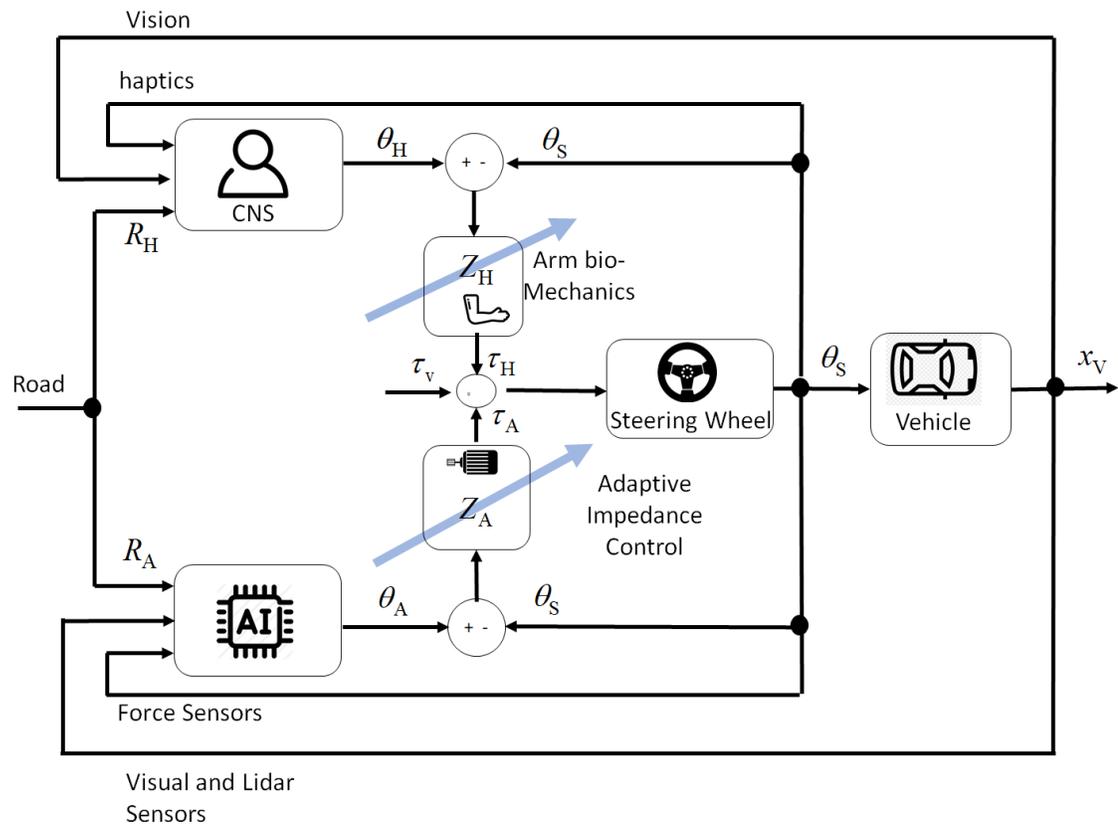


Figure 2.6: A General Model of Haptic Shared Control

The lateral displacement and the yaw angle can be defined as the outputs using;

$$Y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} v(t) \\ \omega(t) \\ y(t) \\ \psi(t) \\ \theta_S(t) \\ \dot{\theta}_S(t) \end{Bmatrix} \quad (2.20)$$

We can write a simplified system model as follow;

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c U_S(t) \\ Y(t) &= C x(t) \end{aligned} \quad (2.21)$$

where

$$\begin{aligned} A_c &= \begin{bmatrix} \frac{-(C_f+C_r)}{U_m} & \frac{-(l_a C_f - l_b C_r)}{U_m} - U & 0 & 0 & \frac{C_f}{mG} & 0 \\ \frac{-(l_a C_f - l_b C_r)}{UI} & \frac{-(l_a^2 C_f + l_b^2 C_r)}{U_m} & 0 & 0 & \frac{l_a C_f}{IG} & 0 \\ 1 & 0 & 0 & U & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{-k_H - k_A}{J_S} & \frac{-b_H - b_A - b_S}{J_S} \end{bmatrix} \\ B_c &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_H}{J_S} & \frac{b_H}{J_S} & \frac{k_A}{J_S} & \frac{b_A}{J_S} \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \\ x &= \left\{ v(t) \quad \omega(t) \quad y(t) \quad \psi(t) \quad \theta_S(t) \quad \dot{\theta}_S(t) \right\}^T \\ U_S &= \left\{ \theta_H(t) \quad \dot{\theta}_H(t) \quad \theta_A(t) \quad \dot{\theta}_A(t) \right\}^T \end{aligned}$$

## CHAPTER 3: Design of Impedance Modulation Controller

This chapter is focused on designing the lower-level controller for the automation system. In particular, this chapter presents an optimal control problem that is solved for modulating the control's impedance to reduce the fight between the human and automation system while ensuring safety. To this end, we designed an adaptive model-predictive controller. Below a description of the proposed controller is given.

### 3.1 Model Predictive Controller

There are various mathematical methods for representing the steering control behavior, for example, model predictive control method, transfer function method, proportional integral derivative method, linear quadratic regulator method, etc. For this research, we have used a model predictive control method (MPC) for describing human and automation's steering control behavior. The main feature of MPC is that it allows the current time period to be optimized while keeping the future time slots into account. Thus MPC can anticipate future events and can take control actions accordingly. We are using a model predictive controller to model the high-level controllers from where we get the motion source for both the agents. In a control problem, the goal of the controller is to calculate the input to the plant such that the plant output follows the desired reference. MPC uses the plant model to make future plant outputs. These future plant outputs are then fed to an optimizer, which ensures that this plant output tracks the desired reference. In our case, we have a vehicle as a plant, and we want the vehicle's trajectory to match the references (human and automation's intents).

A detailed study for the MPC method has been performed by [21]. While perform-

ing a steering control task, the human driver generally previews the road ahead and then accordingly determines the target path to be followed at each time step. This target path is expressed in terms of a sequence of vectors. This sequence of vectors denotes the desired vehicle orientation at future time steps up to the driver's preview horizon. Each vector consists of two elements: the desired lateral displacement and the desired yaw angle. Simultaneously the driver is also receiving the state-feedback of the vehicle, and thus, on the basis of this knowledge, he/she predicts the future vehicle orientation trajectory according to their knowledge of vehicle dynamics.

### 3.2 Adaptive Model Predictive Impedance Controller

In MPC, we have assumed that the Plant model, which comprises of the vehicle dynamics and steering dynamics, is constant. But in reality, it is never constant. In order to change the force applied on the steering wheel, we need to flex our muscles or relax our muscle. Thus we are dynamically changing the stiffness coefficient and the damping coefficient of our hand. This is the adaptive law which we have used to dynamically change the impedance value for both the agents. To present how impedance may evolve in time, we introduce the following dynamic models:

$$\dot{Z}_H(t) = \alpha_H Z_H(t) + \beta_H \Gamma_H(t) \quad (3.1)$$

$$\dot{Z}_A(t) = \alpha_A Z_A(t) + \beta_A \Gamma_A(t) \quad (3.2)$$

where  $Z_H = [B_H \ K_H]^T$  and  $K_H$  and  $B_H$  are the stiffness and damping associated with humans' biomechanics;  $Z_A = [B_A \ K_A]^T$  and  $K_A$  and  $B_A$  are the stiffness and damping associated with the motor's lower-level proportional-derivative controller;  $\Gamma_H = [\Gamma_{bH}(t) \ \Gamma_{kH}(t)]^T$  is the humans control action for modulating his impedance and  $\Gamma_A = [\Gamma_{bA}(t) \ \Gamma_{kA}(t)]^T$  is the automations control input for modulating its impedance

[22]. Additionally,

$$\begin{aligned} \alpha_{\text{H}} &= \begin{bmatrix} \alpha_{\text{bH}} & 0 \\ 0 & \alpha_{\text{kH}} \end{bmatrix}, & \beta_{\text{H}} &= \begin{bmatrix} \beta_{\text{bH}} & 0 \\ 0 & \beta_{\text{kH}} \end{bmatrix} \\ \alpha_{\text{A}} &= \begin{bmatrix} \alpha_{\text{bA}} & 0 \\ 0 & \alpha_{\text{kA}} \end{bmatrix}, & \beta_{\text{A}} &= \begin{bmatrix} \beta_{\text{bA}} & 0 \\ 0 & \beta_{\text{kA}} \end{bmatrix} \end{aligned} \quad (3.3)$$

where  $\{\alpha_{\text{bH}}, \alpha_{\text{kH}}, \alpha_{\text{bA}}, \alpha_{\text{kA}}, \beta_{\text{bH}}, \beta_{\text{kH}}, \beta_{\text{bA}}, \beta_{\text{kA}}\}$  are constant parameters. This formulation captures how impedance evolves in time. Ideally, to determine an optimal behavior for the automation system, optimization should be performed over all control signals of the automation system (i.e.,  $\theta_{\text{A}}, \Gamma_{\text{A}}$ ); However, the focus of this research is to determine  $\Gamma_{\text{A}}$  as means for allocating the level of authority between the driver and the automation system.

### 3.3 Impedance Modulation Controller Design

In this section, we present a predictive controller for modulating the automation impedance such that the assistive behavior of the automation system improves while the safety of the task is also guaranteed.

First, let the discrete-time model of the impedance dynamics (3.1) and (3.2) be

$$Z_{\text{H}}(k+1) = \tilde{\alpha}_{\text{H}}Z_{\text{H}}(k) + \tilde{\beta}_{\text{H}}\Gamma_{\text{H}}(k) \quad (3.4)$$

$$Z_{\text{A}}(k+1) = \tilde{\alpha}_{\text{A}}Z_{\text{A}}(k) + \tilde{\beta}_{\text{A}}\Gamma_{\text{A}}(k) \quad (3.5)$$

where  $\tilde{\alpha}_{\text{A}} = I + T_{\text{s}}\alpha_{\text{A}}, \tilde{\alpha}_{\text{H}} = I + T_{\text{s}}\alpha_{\text{H}}, \tilde{\beta}_{\text{A}} = T_{\text{s}}\beta_{\text{A}},$  and  $\tilde{\beta}_{\text{H}} = T_{\text{s}}\beta_{\text{H}}$  and  $T_{\text{s}}$  is the sampling time. Furthermore, we define

$$\dot{\theta}_i(k) = \frac{\theta_i(k) - \theta_i(k-1)}{T_{\text{s}}} \quad (3.6)$$

where  $i = \{\text{SW}, \text{H}, \text{A}\}$ .

Next, let us define a cost function  $J(k)$  in the form of

$$\min_{\Gamma_A} J(k) = \sum_{j=1}^{N_p} \{ \|\tau_H(k+j) + \tau_A(k+j) - \varepsilon(k+j)\| + \|\tau_H(k+j) - \tau_A(k+j)\| \} \quad (3.7)$$

where  $\tau_H(k) = Z_H[\theta_H(k) - \theta_S(k)]$  and  $\tau_A = Z_A[\theta_A(k) - \theta_S(k)]$ . The first term of the cost function is to ensure safe steering. Specifically, we define  $\varepsilon$  as a minimum required torque that can guarantee the safe maneuver. For this paper, we assume  $\varepsilon$  is known. The second term of the cost function is to minimize the disagreement between a human driver and the automation system. Since the steering angle,  $\theta_S$  and the rate of its changes can be directly measured from the sensor; we simplify the cost function (3.7) into

$$\min_{\Gamma_A} J(k) = \sum_{j=k+1}^{k+N_p} \{ \|\|Z_H(j)\theta_H(j) + Z_A(j)\theta_A(j) - \varepsilon(j)\| + \|Z_H(j)\theta_H(j) - Z_A(j)\theta_A(j)\| \} \quad (3.8)$$

For this research, we assume  $Z_H$  and  $\theta_H$  are known and can be measured. The estimation of human backdrive impedance has a long and rich history in biomechanics and more recent history in the field of haptic rendering. In the context of the adaptive haptic shared control paradigm, parallel with this work, authors are focused on developing a method to obtain a continually-updated estimate of the backdrive impedance of the human driver. With access to an online estimate of the driver's time-varying impedance, as well as the torque on the steering wheel (using a differential torque) sensor, we can estimate the human's intent  $\theta_H$ .

The goal in the cost function is to determine  $\Gamma_A$  such that the cost function  $J$  is

minimized. To this end, we  $Z_A\theta_A$  can be presented as

$$Z_A\theta_A(k) = B_A(k) \left[ \frac{\theta_A(k) - \theta_A(k-1)}{T_s} \right] + K_A(k)\theta_A(k) \quad (3.9)$$

By replacing  $B_A$  and  $k_A$  from Eq. xx, we will have:

$$Z_A\theta_A(k) = \{\Phi(k) + \Psi(k)\} \begin{bmatrix} \theta_A(k) \\ \theta_A(k-1) \end{bmatrix} \quad (3.10)$$

$$\Phi(k) = \tilde{\alpha}_A \begin{bmatrix} \frac{B_A(k-1)}{T_s} + K_A(k-1) & -\frac{B_A(k-1)}{T_s} \end{bmatrix} \quad (3.11)$$

$$\Psi(k) = \tilde{\beta}_A \begin{bmatrix} \frac{\Gamma_{BA}(k)}{T_s} + \Gamma_{KA}(k) & -\frac{\Gamma_{BA}(k)}{T_s} \end{bmatrix} \quad (3.12)$$

The  $\Phi$  and  $\Psi$  represents intrinsic mechanical impedance and control action vectors, respectively. By propagating the automation torque for the next time steps until  $N_p$  step, the  $\Phi$  and  $\Psi$  vectors will move forward in the time. In order to create prediction matrices we can rearrange the  $N_p$  step automation torque vector like following equation:

$$\mathbf{Z}_A \Theta_A = \bar{\Theta} \Omega(\Phi, \Psi) \quad (3.13)$$

where

$$\mathbf{Z}_A \boldsymbol{\Theta}_A = \begin{bmatrix} Z_A(k + N_p)\theta_A(k + N_p) \\ \vdots \\ Z_A(k + 1)\theta_A(k + 1) \\ Z_A(k)\theta_A(k) \end{bmatrix} \quad (3.14)$$

$$\overline{\boldsymbol{\Theta}}^T = \begin{bmatrix} \theta_A(k + N_p) & \cdots & 0 & 0 \\ \theta_A(k + N_p - 1) & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \theta_A(k + 1) & 0 \\ 0 & \cdots & \theta_A(k) & 0 \\ 0 & \cdots & 0 & \theta_A(k) \\ 0 & \cdots & 0 & \theta_A(k - 1) \end{bmatrix} \quad (3.15)$$

$$\boldsymbol{\Omega}(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \begin{bmatrix} \{(\tilde{\alpha}_A)^{N_p}[\Delta(k)] + (\tilde{\alpha}_A)^{N_p-1}\Psi(k + 1) + \dots + \Psi(k + N_p)\}^T \\ \vdots \\ \{\tilde{\alpha}_A[\Delta(k)] + \Psi(k + 1)\}^T \\ \{\Delta(k)\}^T \end{bmatrix} \quad (3.16)$$

where  $\Delta(k) = \Phi(k) + \Psi(k)$ .

According to the second term of (3.8), in the ideal model, the value of  $Z_H\theta_H$  will be equal to  $Z_A\theta_A$ . On the other hand, in the (3.14), the amount of automation control action at a time step  $k$  can be determined by using methods like linear programming (LP), quadratic programming (QP) and least square (LS). As it can be seen in the (3.14), the solution from the optimal solver will give us the summation of  $\Phi$  and  $\Psi$ . In the LP, QP, and LS methods, it is possible to have a negative value. This means, the haptic link on the automation side is disconnected (has zero impedance). In this

research, instead of these methods, the Non-Negative LS method is used, which guaranteed to solve the cost function with a non-negative solution. Furthermore, we used a modified version of NNLS to solve the cost function and reduce the computational burden. In the following section, the MNNLS method is presented.

### 3.3.1 Modified Non-Negative Least Squares Method

Since the impedance is a non-negative parameter, to solve cost function (3.8), the conventional Linear Square algorithm must be rearranged. A non-negative LS is an LS optimization problem which is subjected to non-negativity constraints. A simple, approximate way to implement these constraints is to solve the corresponding unconstrained LS problem and then overwrite any negative values with zeros. However, a challenge associated with overwriting negative values in the LS method is that it may not result in converging to a minimum possible error on successive iterations. To resolve this limitation, there are several mathematically proven methods to solve Non-Negative LS problems [23–25], which impose non-negativity criteria on the solution while minimizing the sum of squared residuals between the data being fitted and their estimates in a real LS sense.

The performance of the NNLS in the large-scale problems which we have large horizon estimation is the main consideration in controller design. Pre-computing the cross product and pseudo-inverses (inverse) matrices are the essential tools for improving the NNLS performance. To resolve this challenge, in this paper, we used a modified NNLS by the following flowchart [24]. The compensatory term in the NNLS is considered in the case of zero impedance from the driver, which means the measurement vector is the summation of the human’s torque and predefined  $\varepsilon$  value.

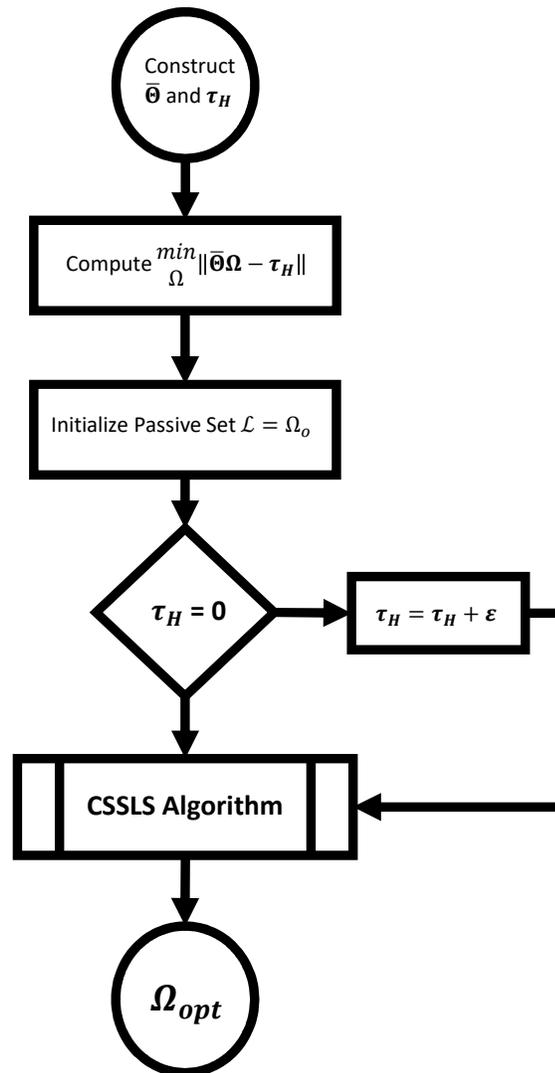


Figure 3.1: MNNLS algorithm

## CHAPTER 4: Developing Game Theoretic Framework for Intent Negotiation

This chapter is mainly focused on the higher-level controller of the haptic shared control paradigms. In particular, we develop a game-theoretic framework to explore the interaction between the human and automation system based on different information patterns available to them.

One of the significant risks associated with haptic shared control framework is when both humans and automation are detecting the same obstacle but decides different paths for avoiding it. In this thesis, we developed models for interactions between the human driver and the automation system for such situations using a game-theoretic framework. In-game theory, players can observe the actions of other players before deciding upon their optimal response. Game theory can be classified based on the mode of the play: whether the two agents are non-cooperating and only thinking about their primary goal or its cooperative.

The players' strategy: An example of the players' strategy is the Nash strategy, where both the agents(driver and automation) only think about themselves, while in a Stackelberg strategy, one agent (human) acts as an initiator while the other agent acts as the conductor. By conductor agent, we mean an agent who considers the optimal input of the initiator to determine its control input. From these two strategies, we can conclude that the two agents in Nash strategy want to be a leader, while in Stackelberg strategy one agent is acting as a leader while the other agent is acting as a follower.

#### 4.1 Non - Cooperative Mode

The equations of motion (2.21) can be described in discrete-time form as

$$x(k+1) = Ax(k) + BU_S(k) \quad (4.1)$$

where  $x(k)$  is the discrete state for time step  $k$ , respectively.

$U_S = \left\{ \theta_H(k) \quad \theta_H(k-1) \quad \theta_A(k) \quad \theta_A(k-1) \right\}^T$  and  $A, B$  are obtained from the discrete bi-linear transformation (2.21) of the corresponding continuous-time matrices  $A_c, B_c$ .

Next, let define the output of the system dynamics (2.21) in a discrete-time form as

$$Y(k) = Cx(k) \quad (4.2)$$

Here,  $C$  is the output matrix which transforms  $x(k)$  into  $Y(k)$ . When we propagate the system to the next time step we get,

$$x(k+2) = Ax(k+1) + BU_S(k+1) = x(k+2) = A^2x(k) + ABU_S(k) + BU_S(k+1) \quad (4.3)$$

The states of the system over the  $N_p$  time steps ahead can be defined and written as [26]

$$\begin{aligned}
& \left\{ \begin{array}{c} x(k+1) \\ x(k+2) \\ x(k+3) \\ \vdots \\ x(k+N_P) \end{array} \right\} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{N_P} \end{bmatrix} x(k) \\
& + \begin{bmatrix} B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ A^2B & AB & B & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N_P-1}B & A^{N_P-2}B & A^{N_P-3}B & \dots & B \end{bmatrix} \left\{ \begin{array}{c} U_S(k) \\ U_S(k+1) \\ U_S(k+2) \\ \vdots \\ U_S(k+N_P-1) \end{array} \right\} \quad (4.4)
\end{aligned}$$

The predicted outputs are obtained as follows;

$$\left\{ \begin{array}{c} Y(k+1) \\ Y(k+2) \\ Y(k+3) \\ \vdots \\ Y(k+N_P) \end{array} \right\} = \begin{bmatrix} C & 0 & 0 & \dots & 0 \\ 0 & C & 0 & \dots & 0 \\ 0 & 0 & C & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & C \end{bmatrix} \left\{ \begin{array}{c} x(k+1) \\ x(k+2) \\ x(k+3) \\ \vdots \\ x(k+N_P) \end{array} \right\} \quad (4.5)$$

Thus substituting equation (4.4) in (4.5), we get;

$$\begin{aligned}
 & \left\{ \begin{array}{c} Y(k+1) \\ Y(k+2) \\ Y(k+3) \\ \vdots \\ Y(k+N_P) \end{array} \right\} = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_P} \end{bmatrix} x(k) \\
 & + \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_P-1}B & CA^{N_P-2}B & CA^{N_P-3}B & \dots & CB \end{bmatrix} \left\{ \begin{array}{c} U_S(k) \\ U_S(k+1) \\ U_S(k+2) \\ \vdots \\ U_S(k+N_P-1) \end{array} \right\} \quad (4.6)
 \end{aligned}$$

Equation (4.6) can be generalized as;

$$y = \Psi x(k) + \mu u_S(k) \quad (4.7)$$

where

$$\begin{aligned}
 y &= \left\{ Y(k+1) \quad Y(k+2) \quad \dots \quad Y(k+N_P) \right\}^T \Psi = \begin{bmatrix} CA & CA^2 & \dots & CA^{N_P} \end{bmatrix} \\
 \mu &= \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_P-1}B & CA^{N_P-2}B & \dots & CB \end{bmatrix} u_S = \left\{ \begin{array}{c} U_S(k) \\ U_S(k+1) \\ \vdots \\ U_S(k+N_P-1) \end{array} \right\}
 \end{aligned}$$

## 4.1.1 Nash Strategy

In a haptic shared control framework, when roles and intents are not communicated, it can be assumed that each agent will try to minimize its cost function without considering the other agents' input into consideration. Specifically, the optimization problem can be written as

$$\begin{aligned}
\min_{U_H} \quad & J_H^{\text{Nash}}(k) = \|Y_H - R_H\|_{Q_H}^2 + \|U_H\|_{R_H}^2 \\
\min_{U_A} \quad & J_A^{\text{Nash}}(k) = \|Y_A - R_A\|_{Q_A}^2 + \|U_A\|_{R_A}^2 \\
s.t. \quad & \begin{cases} x(k+1) = Ax(k) + BU_S(k) \\ Y_H = Cx(k) \\ Y_A = Cx(k) \end{cases} \tag{4.8}
\end{aligned}$$

where

$$\begin{aligned}
R_H &= \begin{bmatrix} y_H & \psi_H \end{bmatrix}^T, \quad R_A = \begin{bmatrix} y_A & \psi_A \end{bmatrix}^T, \\
U_H &= \left\{ \theta_H(k) \quad \theta_H(k-1) \right\}^T \quad \text{and} \quad U_A = \left\{ \theta_A(k) \quad \theta_A(k-1) \right\}^T \\
Q_H, Q_A, R_H, R_A &\text{ are the cost function matrices.}
\end{aligned}$$

We have

$$U_S(k) = \left\{ \theta_H(k) \quad \theta_H(k+1) \quad \theta_A(k) \quad \theta_A(k+1) \right\}^T$$

thus we can write

$$U_S(k) = \left\{ U_H(k) \quad U_A(k) \right\}^T$$

So equation (4.7) can be written as;

$$Y(k) = \Psi x(k) + \mu_H U_H(k) + \mu_A U_A(k) \quad (4.9)$$

To solve the optimization (4.8), we define human's driver and automation's tracking errors as

$$\mathcal{E}_H(k) = T_H - \Psi x(k) - \mu_A U_A(k) \quad (4.10)$$

$$\mathcal{E}_A(k) = T_A - \Psi x(k) - \mu_H U_H(k) \quad (4.11)$$

where  $T_H(k) = [r_H(k+1), \dots, r_H(k+N_p)]^T$ ,  $T_A(k) = [r_A(k+1), \dots, r_A(k+N_p)]^T$ . Substitution of (4.10), (4.40) and (4.9) into (4.8) yields to

$$J_H(k) \triangleq \|\mu_H U_H(k) - \mathcal{E}_H(k)\|_{Q_H}^2 + \|U_H(k)\|_{R_H}^2, \quad (4.12)$$

$$J_A(k) \triangleq \|\mu_A U_A(k) - \mathcal{E}_A(k)\|_{Q_A}^2 + \|U_A(k)\|_{R_A}^2. \quad (4.13)$$

which can be written as [27];

$$J_H(k) = \left\| \begin{bmatrix} S_{Q_H}(\mu_H U_H(k) - \mathcal{E}_H(k)) \\ S_{R_H} U_H(k) \end{bmatrix} \right\|^2 \quad (4.14)$$

$$J_A(k) = \left\| \begin{bmatrix} S_{Q_A}(\mu_A U_A(k) - \mathcal{E}_A(k)) \\ S_{R_A} U_A(k) \end{bmatrix} \right\|^2 \quad (4.15)$$

where  $Q_H = S_{Q_H}^T S_{Q_H}$ ,  $Q_A = S_{Q_A}^T S_{Q_A}$ ,  $R_H = S_{R_H}^T S_{R_H}$  and  $R_A = S_{R_A}^T S_{R_A}$

The optimum value for  $U_H, U_A$  will minimize  $J_H, J_A$  And thus is the solution of;

$$0 = \begin{bmatrix} S_{Q_H}(\mu_H U_H^*(k) - \mathcal{E}_H(k)) \\ S_{R_H} U_H^*(k) \end{bmatrix} \quad (4.16)$$

$$0 = \begin{bmatrix} S_{Q_A}(\mu_A U_A^*(k) - \mathcal{E}_A(k)) \\ S_{R_A} U_A^*(k) \end{bmatrix} \quad (4.17)$$

But we have,  $U_H(k) = \left\{ \theta_H(k) \quad \theta_H(k-1) \right\}^T$  and  $U_A(k) = \left\{ \theta_A(k) \quad \theta_A(k-1) \right\}^T$

Thus, we can rewrite (4.43) as;

$$0 = \begin{bmatrix} S_{Q_H} \mu_H U_H^*(k) + S_{Q_H} \mu_H U_H^*(k-1) - S_{Q_H} \mathcal{E}_H(k) \\ S_{R_H} U_H^*(k) + S_{R_H} U_H^*(k-1) \end{bmatrix} \quad (4.18)$$

$$0 = \begin{bmatrix} S_{Q_A} \mu_A U_A^*(k) + S_{Q_A} \mu_A U_A^*(k-1) - S_{Q_A} \mathcal{E}_A(k) \\ S_{R_A} U_A^*(k) + S_{R_A} U_A^*(k-1) \end{bmatrix} \quad (4.19)$$

or equivalently;

$$\begin{bmatrix} S_{Q_H} \mu_H \\ S_{R_H} \end{bmatrix} U_H^*(k) = \begin{bmatrix} S_{Q_H} \mathcal{E}_H - S_{Q_H} \mu_H U_H^*(k-1) \\ -S_{R_H} U_H(k-1) \end{bmatrix} \\ \begin{bmatrix} S_{Q_A} \mu_A \\ S_{R_A} \end{bmatrix} U_A^*(k) = \begin{bmatrix} S_{Q_A} \mathcal{E}_A - S_{Q_A} \mu_A U_A^*(k-1) \\ -S_{R_A} U_A(k-1) \end{bmatrix} \quad (4.20)$$

The equation (4.45) is similar to  $AX = B$  where  $X$  is unknown while  $A, B$  are known. We solve it using a backslash operator,  $X = A \setminus B$ . This process is called as Least Square method [26].

Thus,

$$U_H^*(k) = K_H \quad (4.21)$$

$$U_A^*(k) = K_A \quad (4.22)$$

where

$$\begin{aligned} K_H &= \begin{bmatrix} S_{Q_H} \mu_H \\ S_{R_H} \end{bmatrix} \setminus \begin{bmatrix} S_{Q_H} \mathcal{E}_H - S_{Q_H} \mu_H U_H^*(k-1) \\ -S_{R_H} U_H(k-1) \end{bmatrix} \\ K_A &= \begin{bmatrix} S_{Q_A} \mu_A \\ S_{R_H} \end{bmatrix} \setminus \begin{bmatrix} S_{Q_A} \mathcal{E}_A - S_{Q_A} \mu_A U_A^*(k-1) \\ -S_{R_A} U_A(k-1) \end{bmatrix}. \end{aligned} \quad (4.23)$$

The human-driver optimal control input depends on the state  $x(k)$ , the desired reference trajectory  $T_H$ , and the automation's steering control  $U_A$  and vice versa. To solve this problem, we employ the convex iteration approach shown in [28], which involves complementing (??) and (4.46) with an auxiliary equation set. In particular,

$$\begin{bmatrix} U_H^{[p+1]}(k) \\ U_A^{[p+1]}(k) \end{bmatrix} = \begin{bmatrix} w_H I & 0 \\ 0 & w_A I \end{bmatrix} \begin{bmatrix} U_H^{*[p]}(k) \\ U_A^{*[p]}(k) \end{bmatrix} + \begin{bmatrix} (1-w_H)I & 0 \\ 0 & (1-w_A)I \end{bmatrix} \begin{bmatrix} U_H^{[p]}(k) \\ U_A^{[p]}(k) \end{bmatrix} \quad (4.24)$$

where  $w_H$  and  $w_A$  are the iteration weights which satisfy  $0 < w_H, w_A < 1$  and  $w_H + w_A = 1$  and  $p$  denotes the step of iteration.

For the first iteration at  $p = 0$  we assume,  $U_H^{[0]}(k) = 0$  and  $U_A^{[0]}(k) = 0$ . Using this

assumption we iterate (4.24) until

$$\begin{bmatrix} U_H^{[\infty]}(k) \\ U_A^{[\infty]}(k) \end{bmatrix} = \begin{bmatrix} w_H I & 0 \\ 0 & w_A I \end{bmatrix} \begin{bmatrix} U_H^{*[\infty]}(k) \\ U_A^{*[\infty]}(k) \end{bmatrix} + \begin{bmatrix} (1 - w_H)I & 0 \\ 0 & (1 - w_A)I \end{bmatrix} \begin{bmatrix} U_H^{[\infty]}(k) \\ U_A^{[\infty]}(k) \end{bmatrix} \quad (4.25)$$

(4.25) has two equations and two unknowns ( $U_H^*$  and  $U_A^*$ ), which can be further solved with very ease. Using this concept of complex iteration approach we can find control input:  $U_H^*$  and  $U_A^*$ .

The control input is the first element of the  $U_H^*$  and  $U_A^*$  sequence. Specifically,

$$u_H(k) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} U_H^* \quad (4.26)$$

$$u_A(k) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} U_A^* \quad (4.27)$$

$u_H$  and  $u_A$  are then further substituted in (4.23) to solve for  $K_H$  and  $K_A$ .

From (4.21) and (4.22), we can see that optimal control input  $U_H^* = K_H$  and  $U_A^* = K_A$ .

#### 4.1.2 Stackelberg Strategy

This section is focused on modeling the interaction within the haptic shared control, assuming one agent acts as an initiator of the task, and the other agent act as a conductor agent. By conductor agent, we mean an agent who considers the optimal input of the initiator to determine its own control input. In this section, we assume the human is an initiator. The automation system considers the "optimal" action of the human driver  $U_H^*(k)$  in planning its actions. Specifically, the optimization problem can be written as

$$\begin{aligned}
\min_{U_H} \quad & J_H^{\text{Stack}}(k) = \|Y_H - R_H\|_{Q_H}^2 + \|U_H\|_{R_H}^2 \\
\text{s.t.} \quad & \begin{cases} x(k+1) = Ax(k) + BU_S(k) \\ Y_H = Cx(k) \end{cases} \\
\min_{U_A} \quad & J_A^{\text{Stack}}(k) = \|Y_A - R_A\|_{Q_A}^2 + \|U_A\|_{R_A}^2 \\
\text{s.t.} \quad & \begin{cases} x(k+1) = Ax(k) + BU_S(k) \\ Y_A = Cx(k) \end{cases}
\end{aligned} \tag{4.28}$$

where

$$\begin{aligned}
R_H &= \begin{bmatrix} y_H & \psi_H \end{bmatrix}^T, \quad R_A = \begin{bmatrix} y_A & \psi_A \end{bmatrix}^T, \\
U_H &= \left\{ \theta_H(k) \quad \theta_H(k-1) \right\}^T \quad \text{and} \quad U_A = \left\{ \theta_A(k) \quad \theta_A(k-1) \right\}^T \\
Q_H, Q_A, R_H, R_A &\text{ are the cost function matrices.}
\end{aligned}$$

We have

$$U_S(k) = \left\{ \theta_H(k) \quad \theta_H(k+1) \quad \theta_A(k) \quad \theta_A(k+1) \right\}^T$$

thus we can write

$$U_S(k) = \left\{ U_H(k) \quad U_A(k) \right\}^T$$

So equation (4.7) can be written as;

$$Y_H(k) = \Psi x(k) + \mu_H U_H(k) + \mu_A U_A(k) \tag{4.29}$$

In Nash's strategy, both the agents are trying to minimize the cost function at the same time because both the agents form a Leader-Leader relationship, but in

Stackelberg, the Human-agent acts as the initiator while the automation agents act as a conductor. Thus the human agent solves its cost function first, and then the automation agent solves its function because it needs to consider the optimal control output of the human agent.

To solve the optimization 4.28, automation's tracking error as

$$\mathcal{E}_H(k) = T_H - \Psi x(k) - \mu_A U_A(k) \quad (4.30)$$

where  $T_H(k) = [r_H(k+1), \dots, r_H(k+N_p)]^T$ . Substitution of (4.40) and (4.39) into (4.28) yields to

$$J_H(k) \triangleq \|\mu_H U_H(k) - \mathcal{E}_H(k)\|_{Q_H}^2 + \|U_H(k)\|_{R_H}^2. \quad (4.31)$$

which can be written as [27];

$$J_H(k) = \left\| \begin{bmatrix} S_{Q_H}(\mu_H U_H(k) - \mathcal{E}_H(k)) \\ S_{R_H} U_H(k) \end{bmatrix} \right\|^2 \quad (4.32)$$

where  $Q_H = S_{Q_H}^T S_{Q_H}$  and  $R_H = S_{R_H}^T S_{R_H}$

The optimum value for  $U_H$  will minimize  $J_H$  And thus is the solution of;

$$0 = \begin{bmatrix} S_{Q_H}(\mu_H U_H^*(k) - \mathcal{E}_H(k)) \\ S_{R_H} U_H^*(k) \end{bmatrix} \quad (4.33)$$

But we have,  $U_H(k) = \left\{ \theta_H(k) \quad \theta_H(k-1) \right\}^T$

Thus, we can rewrite (4.43) as;

$$0 = \begin{bmatrix} S_{Q_H}\mu_H U_H^*(k) + S_{Q_H}\mu_H U_H^*(k-1) - S_{Q_H}\mathcal{E}_H(k) \\ S_{R_H}U_H^*(k) + S_{R_H}U_H^*(k-1) \end{bmatrix} \quad (4.34)$$

or equivalently;

$$\begin{bmatrix} S_{Q_H}\mu_H \\ S_{R_H} \end{bmatrix} U_H^*(k) = \begin{bmatrix} S_{Q_H}\mathcal{E}_H - S_{Q_H}\mu_H U_H^*(k-1) \\ -S_{R_H}U_H(k-1) \end{bmatrix} \quad (4.35)$$

The equation (4.45) is similar to  $AX = B$  where  $X$  is unknown while  $A, B$  are known. We solve it using a backslash operator,  $X = A \setminus B$ . This process is called as Least Square method [26].

Thus,

$$U_H^*(k) = K_H \quad (4.36)$$

where

$$K_H = \begin{bmatrix} S_{Q_H}\mu_H \\ S_{R_H} \end{bmatrix} \setminus \begin{bmatrix} S_{Q_H}\mathcal{E}_H - S_{Q_H}\mu_H U_H^*(k-1) \\ -S_{R_H}U_H(k-1) \end{bmatrix}. \quad (4.37)$$

The human's steering control input depends on the state  $x(k)$ , the desired reference trajectory  $T_H$ , and the automation's steering control  $U_A$ . To solve this problem, we employ the convex iteration approach shown in (4.24). Using this concept we find optimal control output:  $U_H^*$

The optimal control input is the first element of the  $U_H^*$  sequence. Specifically,

$$u_H(k) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} U_H^* \quad (4.38)$$

Now, since we know the value of  $U_H^*$ , we can solve for the optimal automation's output. Thus equation (4.7) becomes;

$$Y_A(k) = \Psi x(k) + \mu_A U_A(k) + \mu_H U_H^*(k) \quad (4.39)$$

To solve the optimization (4.28), we define human's driver and automation's tracking errors as

$$\mathcal{E}_A(k) = T_A - \Psi x(k) - \mu_H U_H^*(k) \quad (4.40)$$

where  $T_H(k) = [r_H(k+1), \dots, r_H(k+N_p)]^T$  Substitution of (4.40) and (4.39) into (4.8) yields to

$$J_A(k) \triangleq \|\mu_A U_A(k) - \mathcal{E}_A(k)\|_{Q_A}^2 + \|U_A(k)\|_{R_A}^2. \quad (4.41)$$

which can be written as [27];

$$J_A(k) = \left\| \begin{bmatrix} S_{Q_A}(\mu_A U_A(k) - \mathcal{E}_A(k)) \\ S_{R_A} U_A(k) \end{bmatrix} \right\|^2 \quad (4.42)$$

where  $Q_A = S_{Q_A}^T S_{Q_A}$  and  $R_A = S_{R_A}^T S_{R_A}$

The optimum value for  $U_A$  will minimize  $J_A$  And thus is the solution of;

$$0 = \begin{bmatrix} S_{Q_A}(\mu_A U_A^*(k) - \mathcal{E}_A(k)) \\ S_{R_A} U_A^*(k) \end{bmatrix} \quad (4.43)$$

But we have,  $U_A(k) = \left\{ \theta_A(k) \quad \theta_A(k-1) \right\}^T$

Thus, we can rewrite (4.43) as;

$$0 = \begin{bmatrix} S_{Q_A} \mu_A U_A^*(k) + S_{Q_A} \mu_A U_A^*(k-1) - S_{Q_A} \mathcal{E}_A(k) \\ S_{R_A} U_A^*(k) + S_{R_A} U_A^*(k-1) \end{bmatrix} \quad (4.44)$$

or equivalently;

$$\begin{bmatrix} S_{Q_A} \mu_A \\ S_{R_A} \end{bmatrix} U_A^*(k) = \begin{bmatrix} S_{Q_A} \mathcal{E}_A - S_{Q_A} \mu_A U_A^*(k-1) \\ -S_{R_A} U_A(k-1) \end{bmatrix} \quad (4.45)$$

The equation (4.45) is similar to  $AX = B$  where  $X$  is unknown while  $A, B$  are known. We solve it using a backslash operator,  $X = A \setminus B$ . This process is called as Least Square method [26].

Thus,

$$U_A^*(k) = K_A \quad (4.46)$$

where

$$K_A = \begin{bmatrix} S_{Q_A} \mu_A \\ S_{R_A} \end{bmatrix} \setminus \begin{bmatrix} S_{Q_A} \mathcal{E}_A - S_{Q_A} \mu_A U_A^*(k-1) \\ -S_{R_A} U_A(k-1) \end{bmatrix}. \quad (4.47)$$

## CHAPTER 5: RESULTS

We consider a scenario in which both human and automation detects an obstacle but choose different paths to avoid it (see Figure 5.1). The desired trajectory for the human driver and the automation system is shown in Figure (5.1). The path for the automation system  $P_A$  can be expressed by the following curve

$$P_A = \begin{cases} 0 & x < l_1 \\ \frac{1}{2}W[\cos(\frac{\pi}{l_2}X - \frac{l_1+l_2}{l_2}\pi) + 1], & l_1 < x < l_1 + l_2 \\ W & l_1 + l_2 < x < l_1 + l_2 + l_3 \\ \frac{1}{2}W[\cos(\frac{\pi}{l_2}X - \frac{l_1+l_2+l_3}{l_2}\pi) + 1], & l_1 + l_2 + l_3 < x < l_1 + 2l_2 + l_3 \\ 0 & x > l_1 + 2l_2 + l_3 \end{cases} \quad (5.1)$$

where  $l_1 = 20$  m,  $l_2 = 20$  m,  $l_3 = 30$  m and  $w = 3$ m and  $P_H = -P_A$ .

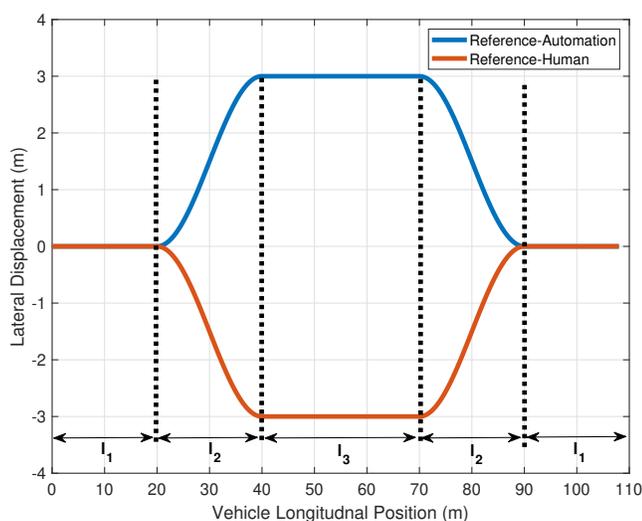


Figure 5.1: Drivers and Automation's target paths

We will be looking at two sets of results, one is without adaptive law (the system model remains constant), and the second one is with adaptive law (the system model changes dynamically). For these two sets of results, we will be looking for a change in the lateral displacement of the vehicle and change in the intent of the human and autonomous agents when we change their impedance, respectively.

## 5.1 Nash Strategy

In Nash Strategy, both the agents form a Leader-Leader relationship. So let's see how it will perform in a non-cooperative paradigm with and without Adaptive Law.

Fig. 5.2, 5.3, 5.5 and 5.6 shows the simulation outcomes for the Nash Strategies. The figure on the top displays the automation and driver target paths and the simulated vehicle lateral displacement in relation to its longitudinal position, while the figure in the middle depicts the driver's and automation's steering angles. The figure in bottom shows how  $K_A$  and  $B_A$  vary with respect to different combinations for  $K_H$  with  $B_H = 0.01$ , thereby displaying the adaptive law. We set horizons to be 10 steps (0.1s). We also select  $q_{\psi,H} = 1, q_{\psi,A} = 1, R_H = 1$  and  $R_A = 1$ .

### 5.1.1 Without Adaptive Law

Under the Nash paradigm, without considering any adaptive impedance controller, the lateral displacement of the vehicle is equal to the average value of the summation of the references for both agents when they have the same impedance characteristics. Fig. 5.2 depicts the same intents for the human driver and the automation system in the case of the same impedance for both agents. It can be observed when  $K_H = 2K_A$  for  $B_H=B_A$ (indicating no adaptive law) the two-game players struggle with each other for the domination of the control of the vehicle, but since  $K_H > 2K_A$ , the human agent dominates the control authority and the vehicle path shifts towards the human reference as seen in Fig. 5.3. Similarly, when  $2K_H = K_A$  for  $B_H=B_A$ (indicating no adaptive law) the two-game players struggle with each other for the domination of

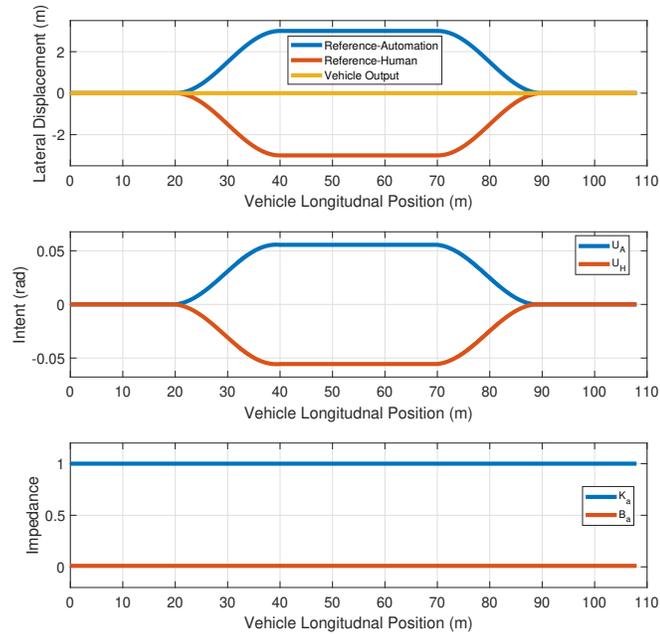


Figure 5.2: Open loop Nash paradigm without adaptive law:

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = K_A$  and  $B_H = B_A$

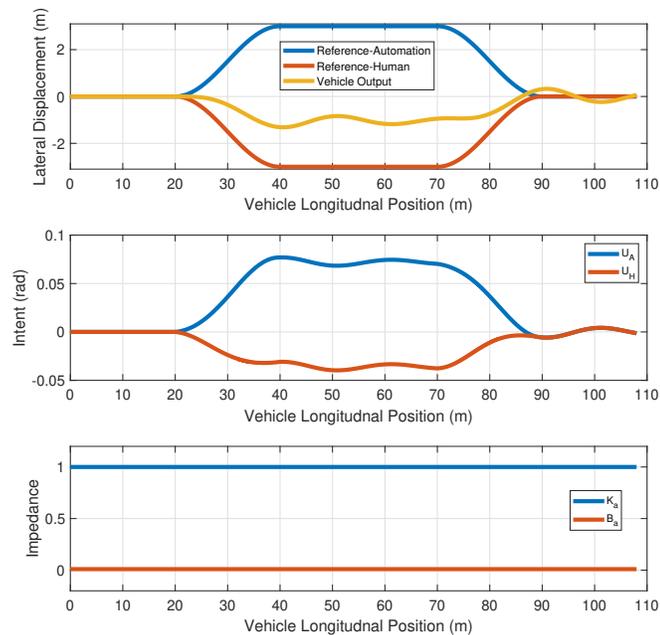


Figure 5.3: Open loop Nash paradigm without adaptive law:

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 2K_A$  and  $B_H = B_A$

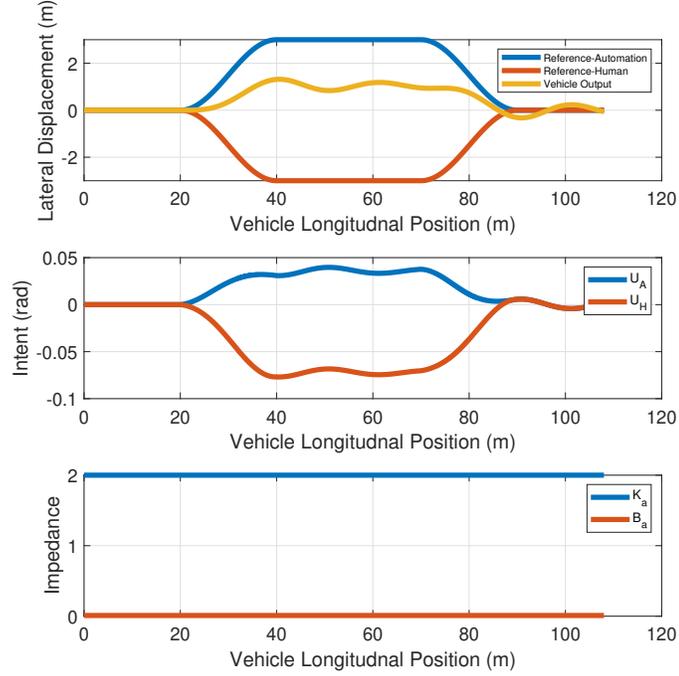


Figure 5.4: Open loop Nash paradigm without adaptive law:

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $2K_H = K_A$  and  $B_H = B_A$

the control of the vehicle but since  $K_A > 2K_H$ , the automation agent dominates the control authority and the vehicle path shifts towards the automation reference as seen in Fig. 5.4

### 5.1.2 With Adaptive Law

In the case of the Adaptive Model Predictive Impedance Controller (AMPIC), the impedance value of the automation system will be regulated based on the human driver behavior and the predefined reference (Fig. 5.5). The dynamic system model will be updated based on the new impedance values of the agents based on the designed supervisory algorithm. The system sample time for the AMPIC subsystem is 10 times smaller than the higher-level controller subsystem in the simulation. The initial condition for the impedance value for the automation is equal to zero. The output of the NNLS algorithm in the AMPIC subsystem is fed to the system model and the Supervisory Model Predictive Controller (SMPC) in the higher-level controllers.

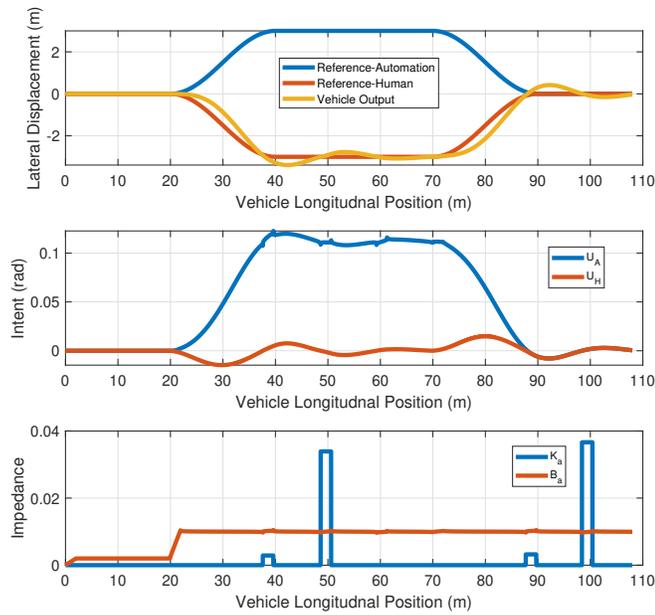


Figure 5.5: Open loop Nash paradigm with adaptive law but constant  $K_H$ :

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 1$  and  $B_H = 0.01$

Also, the AMPIC subsystem has access to the prediction values for the automation horizon prediction for intention and estimated value human intention/impedance. It can be observed that the adaptive law tries to minimize the fight by decreasing its impedance value. The higher-level controller for the automation agent tries to increase the intention value to compensate for the low impedance value. Because of the truncation and round-off errors in the numerical calculations, there is some pulse-like form in the stiffness value. The moving average filter with a wide window length is not applicable for eliminating these jumps in the estimated values for the automation stiffness since it will act as a bypass for the AMPIC subsystem. Fig. 5.5 demonstrates the behavior of the shared control task in the Nash strategy with a supervisory algorithm in the presence of an adaptive model predictive impedance controller when the human driver has constant impedance value. The lateral location of the vehicle follows the human driver reference in this case. By considering the generated intent values for the agents (the middle figure in Fig. 5.5), we can see that the automation's

intent is multiple time bigger than the human driver's intention while the stiffness value of the automation is approximately equal to zero (the bottom figure in Fig. 5.5). It implies that the automation agent understands the fight and reduces its  $K_A$ .

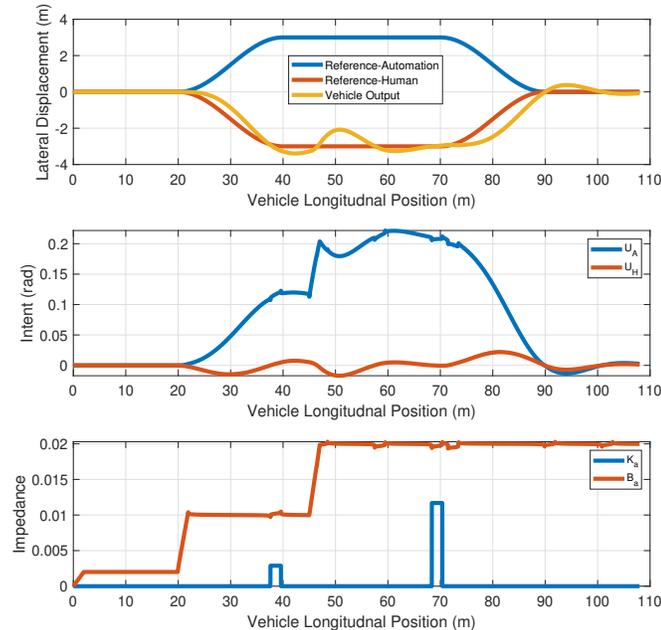


Figure 5.6: Open loop Nash paradigm with adaptive law and varying  $K_H$ :

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H$  is varying and  $B_H = 0.01$

When we use the adaptive law where the  $K_A$  and  $B_A$  vary and  $K_H$  also vary, we see that the automation adapts properly to the change in the  $K_H$ , thereby reducing the fight for control authority in the system. In Fig 5.3, since the stiffness of the human driver is doubled at  $x = 45m$ , the damping coefficient of the automation is increased from 0.01 to 0.02  $N.rad/s$ .

## 5.2 Stackelberg Strategy

In Stackelberg Strategy, the agents form a Leader-Follower relationship. So let's see how it performs in a non-cooperative paradigm with and without Adaptive Law.

Fig. 5.7, 5.8, 5.10 and 5.11 shows the simulation outcomes for the Stackelberg Strategies. The figure on the top displays the automation and driver target paths and

the simulated vehicle lateral displacement in relation to its longitudinal position, while the figure in the middle depicts the driver's and automation's steering angles. The figure in bottom shows how  $K_A$  and  $B_A$  vary with respect to different combinations for  $K_H$  with  $B_H = 0.01$ , thereby displaying the adaptive law. We set horizons to be 10 steps (0.1s). We also select  $q_{\psi,H} = 1, q_{\psi,A} = 1, R_H = 1$  and  $R_A = 1$ .

### 5.2.1 Without Adaptive Law

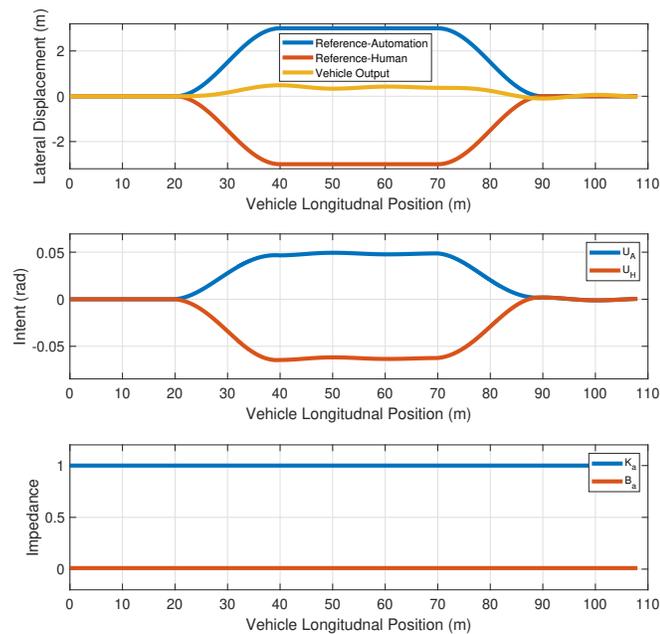


Figure 5.7: Open loop Stackelberg paradigm without adaptive law:

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = K_A$  and  $B_H = B_A$

In Nash Strategy without considering the adaptive impedance controller, the lateral displacement of the vehicle is equal to the average value of the summation of the references for both agents when they have same impedance characteristics, but for Stackelberg Strategy as we can see in Fig 5.7 the lateral displacement of the vehicle is not equal to the average value of the summation of the references for both agents. The lateral displacement of the vehicle tries to follow the leader's reference path, and the extent to which it replicates the leader's reference path depends on the impedance

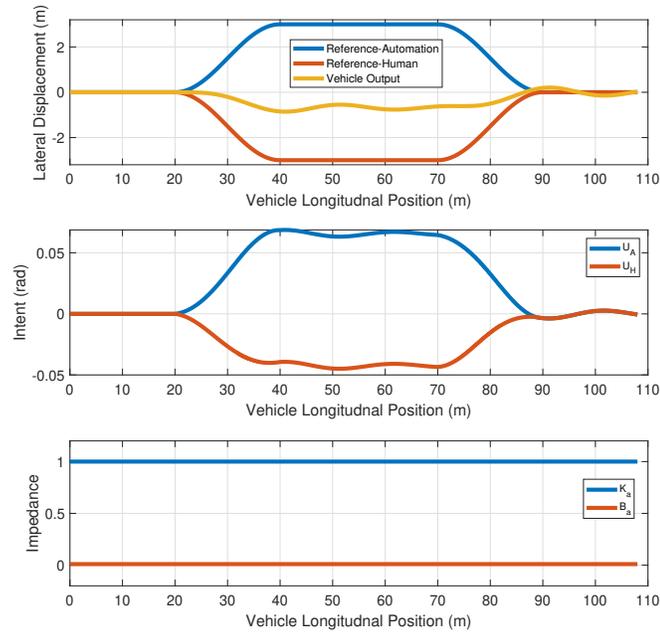


Figure 5.8: Open loop Stackelberg paradigm without adaptive law:

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 2K_A$  and  $B_H = B_A$

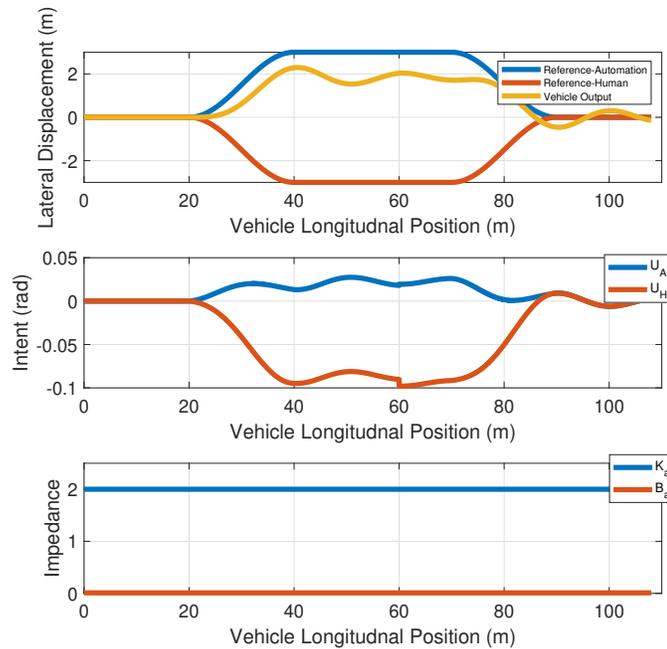


Figure 5.9: Open loop Stackelberg paradigm without adaptive law:

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $2K_H = K_A$  and  $B_H = B_A$

characteristics of both the agents. It can be observed that when  $K_H = 2K_A$  for  $B_H=B_A$  (indicating no adaptive law) that is the follower agent is trying to gain the control of the system as seen in Fig 5.8, the lateral displacement of the vehicle tries to follow the human reference. With  $2K_H = K_A$  for  $B_H=B_A$  (indicating no adaptive law) as seen in Fig 5.9, the Leader agent tries to dominate the system which can be verified by looking at the lateral displacement of the vehicle which approximately replicates the automation reference.

### 5.2.2 With Adaptive Law

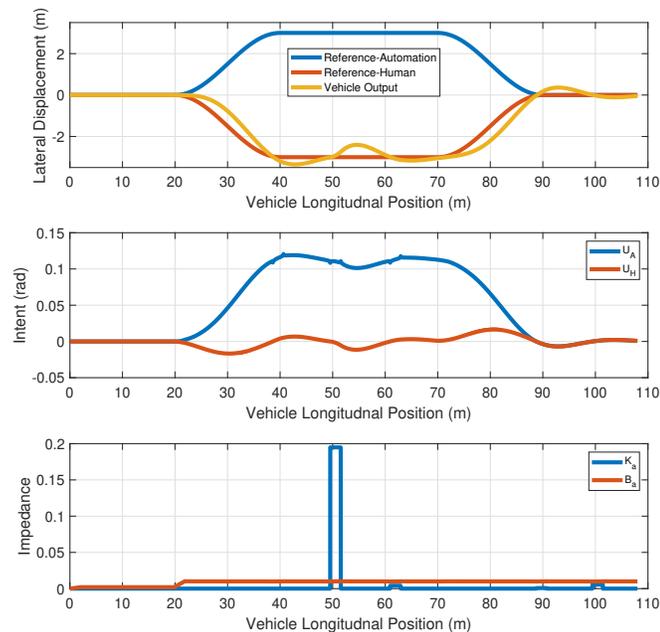


Figure 5.10: Open loop Stackelberg paradigm with adaptive law but constant  $K_H$ : In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H = 1$  and  $B_H = 0.01$

It can be observed that the adaptive law tries to minimize the fight by decreasing its impedance value. The higher-level controller for the automation agent tries to increase the intention value to compensate for the low impedance value. The adaptive impedance controller works efficiently irrespective of different game theory strategies used to model the interactions between two agents. From Fig 5.5, 5.6, 5.10 and 5.11

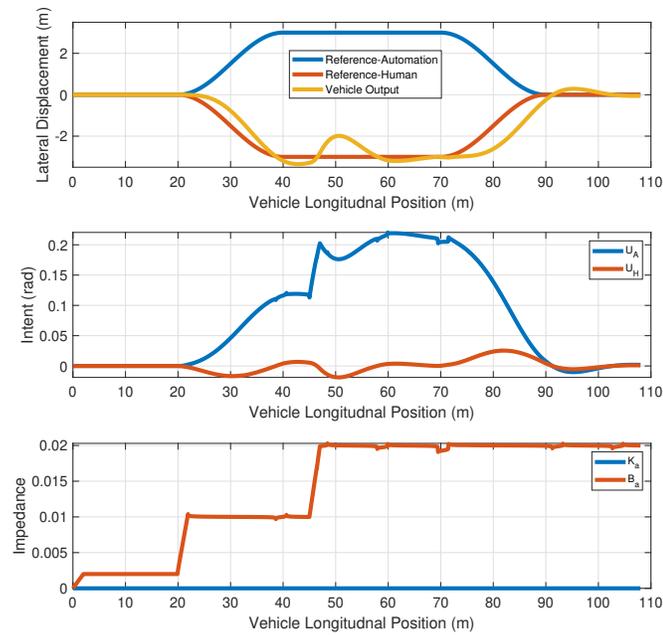


Figure 5.11: Open loop Stackelberg paradigm with adaptive law and varying  $K_H$ :

In (A) the blue and red solid lines represent the automation's and human driver's desired trajectory respectively, while the yellow line represents the vehicle trajectory. (B) represents the intent for the automation and the human driver, (C) represents the value for  $K_A$  and  $B_A$ , when  $K_H$  is varying and  $B_H = 0.01$

we can see that in spite of using different strategies the lateral displacement of the vehicle shows similar pattern for same and varying impedance characteristics.

## CHAPTER 6: CONCLUSIONS and FUTURE WORK

### 6.1 Conclusion

For Nash Strategy, without adaptive law, both the agents do not make any efforts to reduce the fight as both the agents try to form a Leader-Leader relation with each other. If the human impedance is equal to the automation's impedance, the resultant torque acting on the steering wheel will get nullified, and the vehicle will go straight into the obstacle. If either of the agents' impedance is greater than that of the other agent, the vehicle will follow the agents' reference, which has greater impedance. But if we use adaptive law for the Nash strategy, we can see that the automation agent's intention changes with change in the impedance value of the human agent. If the automation agent impedance is less than the human agent's impedance, the automation agent's intention will be greater than that of the human agent. This explains that the automation agent shows a greater intention to reduce the fight between both the agent, thereby converting the non-cooperative interaction to a cooperative one. So if we are using the Nash strategy, it is advisable to use with adaptive law to tackle the non-cooperative situation.

For Stackelberg strategy, without adaptive law, both the agents try to form a leader-follower relation. This helps to convert the non-cooperative situation to a partial cooperative situation. It depends upon the impedance of the human agent. If both the agents have same impedance characteristics, then due to leader-follower relation, the lateral displacement of the vehicle up to some extent follows the Leader agent's reference. If the impedance changes the vehicle's trajectory shifts accordingly towards the agent with greater impedance. With the use of the Adaptive law, we can see that the Stackelberg strategy performs similar to the Nash strategy. The

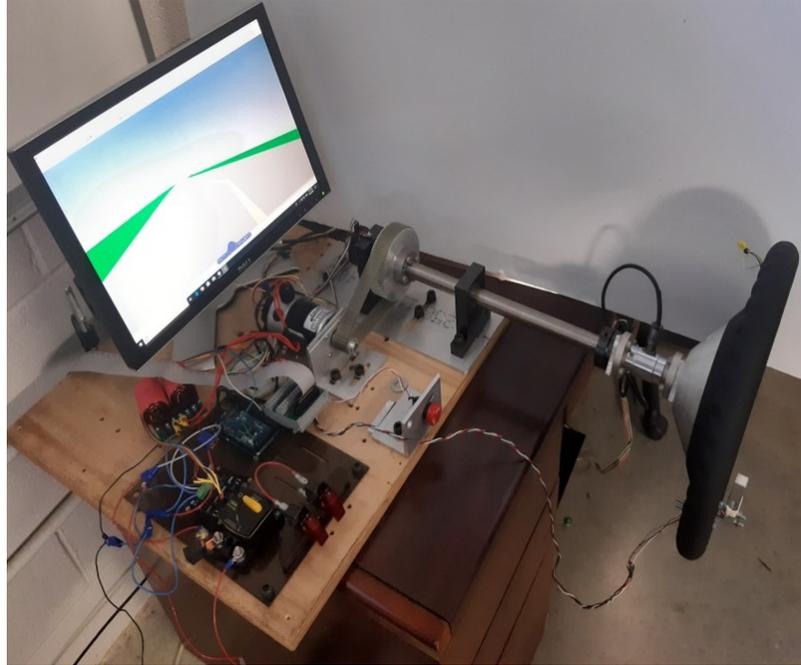


Figure 6.1: A real time simulator

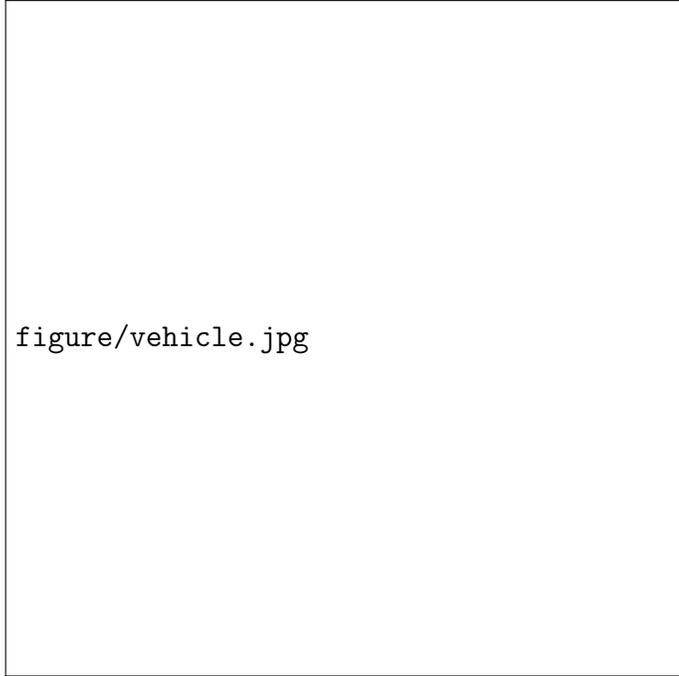
automation agent tries to understand the human agents intention and then reduces its own impedance to convert the non-cooperative strategy to a cooperative one. Thus for non-cooperative strategy without adaptive law, it is better to use the Stackelberg strategy if compared with the Nash strategy. But it is recommended to use Adaptive law, which can work effectively irrespective of the different game theory strategies used to model the interactions between the two agents.

From simulation results, it can be concluded that the Adaptive law when used with any of the strategies, can convert a non-cooperative situation to a cooperative one.

## 6.2 Future Work

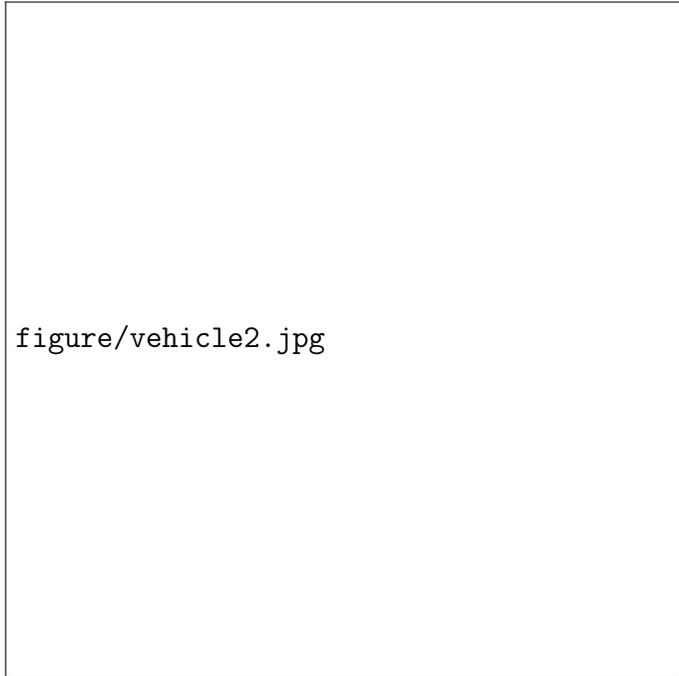
- (1) To perform Numerical simulations for the Cooperative Pareto Strategy and then try different Game theory strategies for different scenarios.
- (2) To experimentally validate the results which we have got for non-cooperative Nash and non-cooperative Stackelberg strategy on a real-time simulator, as shown in 6.1.

- (3) To use haptic shared control framework along with game theory to model a semi-automated vehicle as shown in 6.2.



figure/vehicle.jpg

(a) Human-Automation interface on a Golf vehicle



figure/vehicle2.jpg

(b) Semi-Automated Golf vehicle

Figure 6.2: Semi-Automated Golf vehicle

## REFERENCES

- [1] synopsys, “Levels of autonomy,” 2019.
- [2] D. J. Cole, “Neuromuscular dynamics and steering feel,” tech. rep., Proc. Steering Tech, 2008.
- [3] H.-M. Huang, K. Pavek, B. Novak, J. Albus, and E. Messin, “A framework for autonomy levels for unmanned systems (alfus),” *Proceedings of the AUVSI’s Unmanned Systems North America*, pp. 849–863, 2005.
- [4] J. Anderson, S. Walker and K. Iagnemma, “Experimental Performance Analysis of a Constraint-Based Navigation Framework,” *Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, pp. 1–10, 2014.
- [5] P. G. Griffiths and R. B. Gillespie, “Sharing Control Between Humans and Automation Using Haptic Interface: Primary and Secondary Task Performance Benefits,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 47, pp. 574–590, oct 2005.
- [6] R. Nishimura, T. Wada, and S. Sugiyama, “Haptic Shared Control in Steering Operation Based on Cooperative Status Between a Driver and a Driver Assistance System,” *Journal of Human-Robot Interaction*, vol. 4, pp. 19–37, 2015.
- [7] J. G. Storms and D. M. Tilbury, “Blending of human and obstacle avoidance control for a high speed mobile robot,” in *2014 American Control Conference*, pp. 3488–3493, 2014.
- [8] F. Flemisch, M. Heesen, T. Hesse, J. Kelsch, A. Schieben, and J. Beller, “Towards a dynamic balance between humans and automation: Authority, ability, responsibility and control in shared and cooperative control situations,” *Cognition, Technology and Work*, vol. 14, no. 1, pp. 3–18, 2012.
- [9] D. A. Abbink, M. Mulder, and E. R. Boer, “Haptic shared control: smoothly shifting control authority?,” *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.
- [10] C. C. MacAdam, “An optimal preview control for linear systems,” 1980.
- [11] C. C. MacAdam, “Application of an optimal preview control for simulation of closed-loop automobile driving,” *IEEE Transactions on systems, man, and cybernetics*, vol. 11, no. 6, pp. 393–399, 1981.
- [12] H. Peng, “Evaluation of driver assistance systems-a human centered approach,” in *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, 2002.

- [13] A. J. Pick and D. J. Cole, "Dynamic properties of a driver's arms holding a steering wheel," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 221, no. 12, pp. 1475–1486, 2007.
- [14] H. R. Hajian AZ, "Identification of the Mechanical Impedance at the Human Finger Tip.," *ASME Journal of BioMedical Engineering*, vol. 119, pp. 109–114, 1997.
- [15] C. J. Hasser and M. R. Cutkosky, "System identification of the human hand grasping a haptic knob," in *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, no. 171–180, 2002.
- [16] B. Yu, R. B. Gillespie, J. S. Freudenberg, and J. A. Cook, "Identification of human feedforward control in grasp and twist tasks," in *American Control Conference*, pp. 2833–2838, 2014.
- [17] D. I. Katzourakis, D. A. Abbink, E. Velenis, E. Holweg, and R. Happee, "Driver's arms' time-variant neuromuscular admittance during real car test-track driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, pp. 221–230, 2014.
- [18] J. Bendtsen and K. Trangbaek, "Closed-loop identification for control of linear parameter varying systems," *Asian Journal of Control*, vol. 16, no. 1, pp. 40–49, 2014.
- [19] M. Olivari, F. M. Nieuwenhuizen, H. H. Bülthoff, and L. Pollini, "Identifying time-varying neuromuscular response: A recursive least-squares algorithm with pseudoinverse," in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pp. 3079–3085, IEEE, 2015.
- [20] A. Pronker, D. Abbink, M. Van Paassen, and M. Mulder, "Estimating driver time-varying neuromuscular admittance through lpv model and grip force," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14916–14921, 2017.
- [21] X. Ji, Y. Liu, X. Na, and Y. Liu, "Research on interactive steering control strategy between driver and afs in different game equilibrium strategies and information patterns," *Vehicle system dynamics*, vol. 56, no. 9, pp. 1344–1374, 2018.
- [22] V. Izadi and A. Ghasemi, "Determination of roles and interaction modes in a haptic shared control framework," in *Proceedings of the ASMA Dynamic Systems and Control Conference in Park City, Utah*, pp. 1–8, 2019.
- [23] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 11, no. 5, pp. 393–401, 1997.

- [24] M. H. Van Benthem and M. R. Keenan, “Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems,” *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 10, pp. 441–450, 2004.
- [25] J. Kim, Y. He, and H. Park, “Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework,” *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, 2014.
- [26] X. Na and D. J. Cole, “Linear quadratic game and non-cooperative predictive methods for potential application to modelling driver–afs interactive steering control,” *Vehicle System Dynamics*, vol. 51, no. 2, pp. 165–198, 2013.
- [27] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.
- [28] D. Mayne and J. Rawlings, “Model predictive control: theory and design,” *Madison, WI: Nob Hill Publishing, LCC*, 2009.