

The Pennsylvania State University
The Graduate School
Department of Mechanical and Nuclear Engineering

**IDENTIFICATION OF STABILITY THRESHOLDS IN TIME-DELAYED
VEHICLE TELEOPERATION**

A Thesis in
Mechanical Engineering
by
Kevin S. Swanson

© 2013 Kevin S. Swanson

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2013

The thesis of Kevin Swanson was reviewed and approved* by the following:

Sean Brennan
Associate Professor, Department of Mechanical Engineering
Thesis Advisor

H. Joseph Sommer
Professor, Department of Mechanical Engineering
Committee Member

Karen A. Thole
Professor, Department of Mechanical Engineering
Head of the Department of Mechanical Engineering

*Signatures are on file in the Graduate School

ABSTRACT

This thesis provides an analysis of vehicle stability in time-delayed teleoperation. As teleoperation continues to grow as a field, it is important to ensure that the human operator of teleoperated vehicles is provided as much feedback as possible to maintain safe and high performance operation. While it is unrealistic to believe that the operator will have feedback in a fully immersive environment, it is possible to provide the operator with the feedback cues primarily utilized during normal human driving. To determine how drivers use visual cues, a series of simulations and physical tests were run to provide both time domain and frequency domain analysis of vehicle stability under systems with either limited preview or increasing time delays. This research provides evidence of how drivers use previewed visual cues to control a vehicle along a path, along with how much delay a driver is able to withstand in the system. The results show that human drivers use the nearest visual information possible during path tracking and can control a vehicle up to about 0.35s of delay at 10m/s. This information is useful in the design of teleoperated systems in which high time delays are common.

TABLE OF CONTENTS

List of Figures.....	vi
List of Tables	ix
Acknowledgements.....	x
Chapter 1 Introduction	1
1.1 Telerobotics.....	1
1.2 Driving simulators and HIL applications	4
1.3 Motivation and goal.....	7
1.4 Thesis organization.....	7
Chapter 2 Review of Existing Technologies and Research	9
2.1 Review of vehicle and robot teleoperation.....	9
2.1.1 The human role in telerobotics	9
2.1.2 Haptics.....	14
2.1.3 Vehicle based teleoperation.....	18
2.1.4 Transparency in driving simulators	21
2.1.5 Further areas of research	22
2.2 Latency issues and testing procedures in telerobotics.....	23
2.2.1 Internet based teleoperation.....	23
2.2.2 Strategies for controlling system latency	25
2.2.3 Common testing procedures.....	30
Chapter 3 Software Environment.....	34
3.1 Overview of ROS	34
3.2 Justification of use.....	35
3.3 ROS to ROS communication latency	37
Chapter 4 Testing Procedure and Platform	39
4.1 Test vehicle.....	39
4.1.1 Physical vehicle description	39
4.1.2 Hardware layout	40
4.1.3 Hardware latency and bandwidth analysis	44

4.2 Test vehicle modeling	49
4.2.1 Bicycle model.....	49
4.2.2 Simulation development.....	52
4.2.3 Simulating a human driver	56
4.3 Testing parameters	59
4.3.1 Test descriptions.....	59
4.3.2 Physical testing conditions	62
Chapter 5 Vehicle Simulation with Varying Preview Time and System Delays.....	64
5.1 Test 1 – MacAdam simulations with varying preview time.....	64
5.1.1 Test setup and results	64
5.1.2 Discussion	71
5.2 Test 2 – MacAdam simulations with varying time delays	73
5.2.1 Test setup and results	73
5.2.2 Discussion	75
Chapter 6 Physical Testing with Varying Preview Time and System Delays	78
6.1 Test 3 – Physical tests with varying preview time	78
6.1.1 Test setup.....	78
6.1.2 Results	80
6.1.3 Discussion	83
6.2 Test 4 – Physical tests with varying delay.....	85
6.2.1 Test setup.....	85
6.2.2 Results	87
6.2.3 Discussion	92
Chapter 7 Conclusions and Future Work.....	97
7.1 Conclusions	97
7.2 Future Work	99
References.....	102
Appendix A – Derivation of the Bicycle Model with the Motor Model.....	106
Appendix B – Derivation of the Feedback Equations for the MacAdam Controller.....	108

LIST OF FIGURES

Figure 1-1. Telerobotic device for bomb disposal [1]	2
Figure 1-2. Basic master-slave feedback loop	4
Figure 1-3. Interior (left) and motion platform (right) of Toyota’s driving simulator [6]	5
Figure 2-1. Operation of a master-slave system	10
Figure 2-2. Operation of a supervisor-subordinate system	11
Figure 2-3. Operation of a partner-partner system.....	12
Figure 2-4. Operation of a teacher-learner system.....	13
Figure 2-5. Operation of a fully autonomous system.....	14
Figure 2-6. Plot of force feedback laws used by Toffin [21].....	16
Figure 2-7. Force feedback responses with various materials [25].....	18
Figure 2-8. Predator drone UAV [27].....	19
Figure 2-9. ROV used by Chouiten [26].....	20
Figure 2-10. Lunokhod 1 moon rover [3]	21
Figure 2-11. Comparison of latency and bandwidth for three communication protocols.....	25
Figure 2-12. Round trip response times for 84 byte data packets [46]	26
Figure 2-13. RMS error of pendulum position for various sampling rates [46]	26
Figure 2-14. Two-state Markov chain between open and closed loop system [48].....	27
Figure 2-15. Traditional time-based control (a) and event-based control (b) [49]	29
Figure 2-16. Comparison of time-based response (left) to event-based response (right) [49].....	30
Figure 2-17. Example command vs. time plot for operator and robot [50]	31
Figure 2-18. Bode plot showing improved high frequency response [22].....	31
Figure 3-1. Basic ROS node diagram	35
Figure 3-2. Latency for a wired ROS to ROS connection	38
Figure 3-3. Latency for a wireless ROS to ROS connection	38

Figure 4-1. 1989 GMC 2500 “Big Red”	40
Figure 4-2. Novatel SPAN RTK GPS.....	41
Figure 4-3. PointGrey Research Firefly MV USB 2.0 camera	41
Figure 4-4. Logitech G25 racing wheel	42
Figure 4-5. Aerotech rotary stage mounted on the steering column	43
Figure 4-6. Hardware diagram for the test vehicle	44
Figure 4-7. Steering motor response for a commanded 90° and 180° wheel angle	46
Figure 4-8. Simulink diagram for the steering motor	47
Figure 4-9. Comparison of simulated and real commanded 90° wheel angle	48
Figure 4-10. Comparison of simulated and real commanded 180° wheel angle	49
Figure 4-11. Planar Bicycle Model.....	50
Figure 4-12. Simulink diagram of the bicycle model with steering motor response	52
Figure 4-13. Local vehicle to global coordinate conversion.....	53
Figure 4-14. Simulated lane change maneuver for the truck at 20m/s.....	54
Figure 4-15. Simulated vs. real response for a double lane change at 16m/s [56].....	54
Figure 4-16. Steering command vs. steering motor response	55
Figure 4-17. X position vs. Y position during a double lane change with motor response at 20m/s.....	56
Figure 4-18. Graphical representation of the MacAdam model for lateral error	57
Figure 4-19. Lateral position vs. time for a lane change at 20m/s with MacAdam controller	59
Figure 4-20. In-vehicle driving simulator	61
Figure 4-21. View from driving simulator with limited preview time	62
Figure 4-22. Circular path in the testing area at LTI	63
Figure 5-1. Comparison of block diagram based motor model to state space based motor model.....	66

Figure 5-2. Bode plot of the closed-loop system with 0.3 seconds of driver preview without actuator.....	68
Figure 5-3. Bode plot of the closed-loop system with 0.3 seconds of driver preview with actuator.....	68
Figure 5-4. Example results for a lane-change maneuver for MacAdam controller for various preview times	71
Figure 5-5. Example results for a lane-change maneuver for MacAdam controller for various delays.....	74
Figure 5-6. Lane change maneuver at 25m/s with 0.15s delay	77
Figure 6-1. Comparison of full camera view to limited camera view	79
Figure 6-2. Real vehicle path vs. guide path for a 25m preview distance	81
Figure 6-3. Root mean square error vs. preview time.....	82
Figure 6-4. Actual command vs. delayed command in a 0.2s one-way delay	86
Figure 6-5. Real vehicle path vs. guide path for a 0.3s round-trip delay	88
Figure 6-6. Real vehicle path vs. guide path for a 0.4s round-trip delay	88
Figure 6-7. Vehicle velocity during 0.55s delay	89
Figure 6-8. Vehicle velocity during 0.45s delay	90
Figure 6-9. Root mean square error vs. velocity for various time delays	91
Figure 6-10. Zoomed plot of root mean square error vs. velocity for various time delays.....	91
Figure 6-11. Lane change maneuver at 7.53m/s with 0.55s delay	93
Figure 6-12. Lane change maneuver at 7.4m/s with 0.45s delay	94
Figure 6-13. Lane change maneuver at 6.8m/s with 0.45s delay	94
Figure 6-14. Lane change maneuver at 6.8m/s with 0.45s delay, but at 0.43s preview.....	95

LIST OF TABLES

Table 2–1. RMS variations obtained in driving simulation experiment [21].....	16
Table 2–2. System performance data [52]	32
Table 3–1. Results of ROS to ROS latency tests	37
Table 4–1. Vehicle parameters for test vehicle.....	39
Table 4–2. Data refresh rates for test vehicle hardware.....	45
Table 4–3. Wired and wireless latency for PointGrey camera.....	45
Table 4–4. Bandwidth of vehicle hardware	45
Table 4–5. Steering motor eigenvalues.....	48
Table 4–6. List of bicycle model properties	51
Table 5–1. Phase margin for simulated preview time test without steering motor model	69
Table 5–2. Phase margin for simulated preview time test with steering motor model	70
Table 5–3. Vehicle stability condition for various time delays	75
Table 6–1. Results from varying preview time experiment.....	83
Table 6–2. Results from varying delay experiment	92

ACKNOWLEDGEMENTS

I primarily want to thank Dr. Brennan and the newly certified Dr. Alex Brown for their steadfast help throughout my 3+ years in the Intelligent Systems and Vehicles group here at Penn State. They have both provided significant help and guidance for me throughout my time here and I do not feel that my education would have truly been complete had it not been for their influence. Alex has been especially helpful in the past year while Dr. Brennan was away; serving as a surrogate adviser and helping me accomplish my thesis goals while still working on his own.

I also want to thank Jesse Pentzer and Bobby Leary, along with the rest of our research group for their assistance in testing and data processing. Jesse was always ready to throw down whatever he was doing to help and his general knowledge has saved me on many occasions. Bobby was always willing to help with testing, even when it took out an entire day of his own work. Without their aid, this thesis would not have been completed.

Finally, I must thank my family for all of the support they have given me my entire life. Without their enduring support I would not be who I am today.

CHAPTER 1

INTRODUCTION

This thesis presents an investigation of how human drivers use feedback to guide and control teleoperated vehicles, along with the design of an intervention architecture by which the remote vehicle can prevent instabilities in the event of delay or loss of feedback. The purpose of this work is to guide the design of vehicle teleoperation systems by identifying specific feedback cues that are most important for driver perception and stable vehicle operation. The motivation of this work is to develop the capability for remote driving of a robotic ground vehicle, such as could occur in space exploration, mining operations, hazardous chemical cleanup, or even defense operations. These examples share the operational attributes of expensive and highly challenging situations that are likely unable to provide all feedback cues that exist in the remote environment to a local driver.

1.1 Telerobotics

Telerobotics, one of the oldest subcategories in the field of robotics, refers to the control of a robot through a human controller at some physically separated distance [1] [2]. The distance separating the controller from the robot is arbitrary and can vary from being in the same room to being on a different planet. Teleoperation has been implemented since the 1940s when it was primarily used by Raymond C. Goertz for nuclear research to handle radioactive material while behind a shielded safety barrier [1]. Today, the purpose of teleoperation has expanded beyond the scope of just safety to include uses such as reaching remote environments like space or the deep

ocean, working with extremely large or small objects like in surgical operations, and even just for use as a toy like an RC car, among many others [1].

The teleoperated vehicle has been described as “...a fully-mobile, physical proximity (a ‘real-world avatar’) for the operator” [3]. Teleoperation development has accelerated in recent years due to video and computing compression improvements, and the use of high quality Light Detection and Ranging (LiDAR) optical remote sensing technology [4]. The means of communication between the operator and robot has also changed with the emergence of new technologies. Communication protocols can be as simple as a radio signal like in an RC car, or a complex system that utilizes the most modern data capacity and transmission capabilities of the Internet [1]. Other options such as Bluetooth communication exist, but Wi-Fi and wired Internet communication will be the primary focus of this project.



Figure 1-1. Telerobotic device for bomb disposal [1]

There are a number of levels of control that can be achieved in telerobotics. The most simple is called *direct control* or *manual control* [1]. At this level, the human user controls the

robot directly without any automated assistance and little feedback. An example of this is shown in Figure 1-1 with a military bomb disposal robot that is directly controlled by a human user who receives video feedback to guide the robot while in a protected stance. On the other end of the spectrum is *supervisory control* [1]. At this level, the robot system has a very high level of automation and the human user only takes control of the system in case of a procedural error or safety hazard. Most teleoperation systems exist at some medium between these two extremes.

In all instances of teleoperation short of full autonomy there is some ability for the user to take direct control of the robot through some interface. This interface is traditionally referred to as the *master-slave interface* where the master is the human user and the slave is the robot [1]. A basic feedback loop is shown in Figure 1-2 where the master sends a command through a controller to the slave. The slave then performs the commanded task and sends feedback to the master. In a good master-slave system the slave will provide some amount of haptic feedback to the user, and in the best systems the slave will provide haptic feedback that is indistinguishable from direct interaction. This feedback could range from visual information like a video feed from the robot to a force feedback on the master's control device. The end goal of this feedback is *telepresence*, which is the ability of the master to not only control the robot, but to fully perceive the environment in which the robot is operating [1] [2]. Full telepresence is extremely difficult to achieve; most remote environments can include cues for all senses including ones that are difficult to simulate, for example touch or smell.

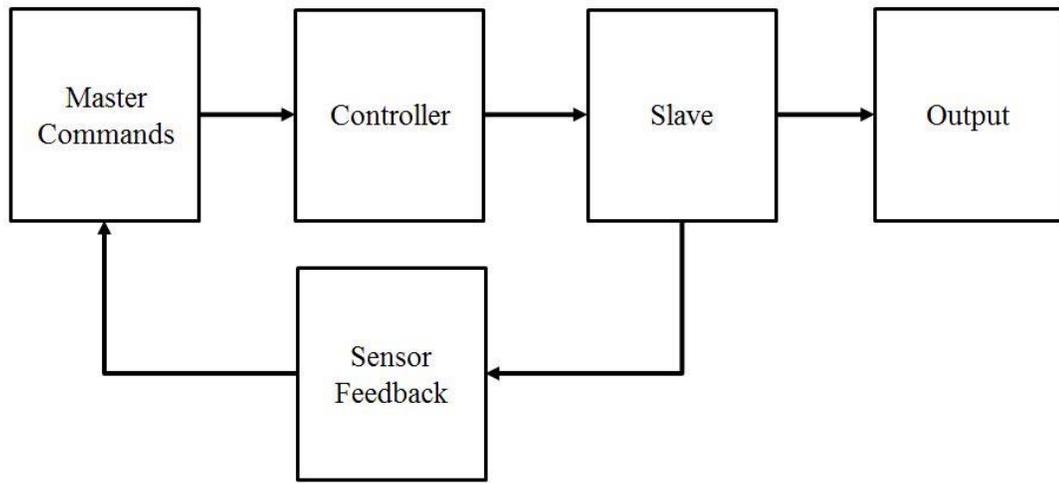


Figure 1-2. Basic master-slave feedback loop

It is even challenging to measure degrees of telepresence. A commonly used metric is one of *transparency* which is the human operator's ability to distinguish between their operating environment and the actual robot's environment [2]. This topic is discussed more fully in Chapter 2's review of literature on system latency.

1.2 Driving simulators and HIL applications

Driving simulators are of high interest to this work because of their ability to provide a large variety of feedback to human drivers from visual cues to force feedback on steering wheels and vehicle motion. Specifically, hardware-in-the-loop (HIL) simulators are of special interest because of their ability to interact with real hardware including full vehicles. This section provides a background on HIL driving simulators with a focus on their ability to create a high level of transparency to both the human driver, and to any physical hardware interacting with the driving software in order to emulate a real vehicle interface.

Driving simulators have been an object of increasing focus in recent years because of their many applications including vehicle/traffic control testing, testing of driver performance and driver training under various circumstances, and development of realistic driving games [5]. They come in many designs from small, home-built, open-source simulators to large scale, multi-million dollar, commercial simulators like the one at Toyota's Higashifuji Technical Center shown in Figure 1-3.



Figure 1-3. Interior (left) and motion platform (right) of Toyota's driving simulator [6]

Controlled driving environments such as test tracks and driving simulators share the common purpose of evaluating human performance with minimal driving risk. Test track facilities allow the driver to interact with the road, providing a higher sense of realism but reducing the reproducibility of a traffic driving environment; in contrast, driving simulators allow high reproducibility of driving scenarios at the expense of realism [7]. This similarity between test tracks and driving simulators begs the question: what tools are needed to facilitate a combination of both experiences to achieve the goal of telepresence discussed in Section 1.1?

To obtain realism, one assumption is that the simulator driver perceives risk similar to that of driving an actual vehicle. According to Risk Homeostasis Theory, drivers may adapt more risky behavior if the perceived consequences of the behavior are reduced. An example of this

effect would be how modern drivers are more likely to drive in a snow storm if their vehicle has four wheel drive. In driving simulators, because a driver's action will not result in an accident, they may not behave in a simulator as they do on the road [8] unless exposed to particularly risky driving behaviors from other drivers rather than simply simulated traffic [9]. One means of improving a driver's perceived realism is to use simulators as interfaces for humans to drive real vehicles, with real life consequences in terms of external safety and vehicle motion. Preliminary testing on rolling roadway simulators in which a driver remotely operates a physical vehicle driving on a treadmill nearby the operator showed that drivers appeared more cautious when driving real vehicles than if they were driving virtual vehicles. This suggests that drivers are far more cautious driving actual systems knowing that physical damage will take place in the presence of poor driving choice, even if their own safety isn't involved [10] [11] [12]. This is a key result supported by Risk Homeostasis theory [8].

Recently, simulators have moved beyond basic driver testing and training, and now many simulators are used for specialized purposes such as controlling hardware, allowing multiple drivers-in-the-loop, or interacting with real vehicles. Despite these simulation advances, automotive research using live HIL in combination with Human-in-the-Loop testing, sometimes called H²iL, has been underutilized [13]. Since all simulators are HIL testing systems, with the human being the "hardware" component, there is not a clear dividing line between human-HIL simulators and fully HIL simulators. In this thesis, the mention of a HIL simulator refers to a system where both the driver and another vehicle hardware component are interacting with each other.

There are many advantages to HIL testing versus simulation or in-vehicle testing. According to Fathy [14], these advantages include higher fidelity, faster simulation speed than purely virtual systems, and greater comprehensiveness than purely physical systems. To achieve

these advantages, the system must bring together many key factors including: real-time operating systems, high-fidelity dynamic models, high bandwidth networking, and low-latency hardware/software integration. This paper will aim to meet all of these goals in the development of the HIL simulator.

1.3 Motivation and goal

As discussed earlier in this chapter, full telepresence can be extremely difficult and costly to achieve. While a full creation of the remote environment would be an extremely powerful tool, it is not completely necessary as humans do not rely on all of their senses equally when operating a vehicle. This work seeks to obtain an understanding of which feedback cues human drivers rely on most heavily, and in particular the timing of those cues. This information is crucial to guide the design of vehicle teleoperation systems to achieve a higher level of telepresence with a lower total cost.

The specific goal of this thesis is to limit a driver's feedback cues in a safe testing environment and monitor the response of a teleoperated vehicle in order to determine which are most heavily relied upon during teleoperation.

1.4 Thesis organization

The remainder of this paper will be organized as follows: Chapter 2 provides an extensive review of past research surrounding vehicle teleoperation and transparency in HIL simulators, along with a discussion on common testing methods for validating system latency. Chapter 3 discusses the software platform, Robot Operating System (ROS), which is primarily utilized in this paper. Chapter 4 presents the testing procedures including the physical vehicle used in

testing. Chapter 5 presents simulated experiments with either varying preview time for a constant delay or varying delay for a constant preview time. Chapter 6 presents physical experiments to verify the simulation results. A conclusions chapter then provides a summary and conclusion of all work performed.

CHAPTER 2

REVIEW OF EXISTING TECHNOLOGIES AND RESEARCH

This chapter provides a review of past work performed in the areas of vehicle and robot teleoperation and hardware-in-the-loop simulations, along with currently available technologies known to be presently fielded for ground vehicle operation. Additionally, it covers the primary testing procedures to find system latency and prove fidelity that have been utilized in past work.

2.1 Review of vehicle and robot teleoperation

2.1.1 The human role in telerobotics

Aracil [15] states that the human role in telerobotics has traditionally been to act as the operator of a robot driving in some remote environment. In its simplest form, the operator takes control by pressing some button or moving a lever, observes the motion in the remote environment, and then responds accordingly. This type of control is typically referred to as *master-slave* control. One example of this control architecture is force-velocity control in which the operator sends a force signal to the robot, and the robot responds with its local velocity [16]. Figure 2-1 shows the full operation of a master-slave system. At its most basic, a command is sent from an operator's controller through a command processor to an actuator controller on a robot through some communication protocol, which in turn moves an actuator. The remote device then sends sensor data back through the sensor processor to a feedback processor at the operator through some communication protocol, which then provides some amount of haptic feedback. The local controller in each the operator and remote environment are not standard to all master-slave systems, but it used can provide some amount of autonomy through a dynamic model in the

case when communication between the operator and remote environment is dramatically slower than the actuator movement.

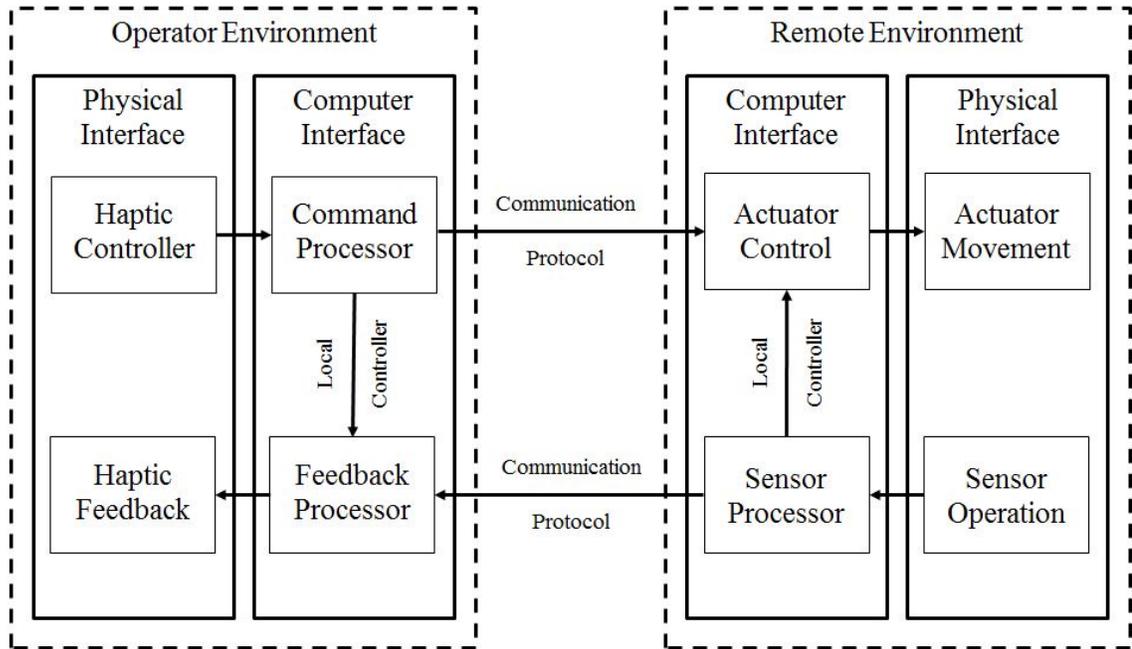


Figure 2-1. Operation of a master-slave system

Aside from master-slave control, there are other types of control including *supervisor-subordinate*, *partner-partner*, *teacher-learner*, and *full autonomy* [17]. In all of these control methods the robot has some amount of autonomy. In supervisor-subordinate control, shown in Figure 2-2, the robot is automated to perform a pre-planned sequence of tasks. Primary computing occurs in the remote environment's task algorithm, but feedback is still sent back to the operator's environment to enable potential overrides. In the figure, dotted lines are provided to indicate actions that do not occur continuously, but on an "as needed" basis. This method is used by Manuaba [18] to control to movement of a block through an environment laden with obstacles using both a direct motion method and a queuing method.

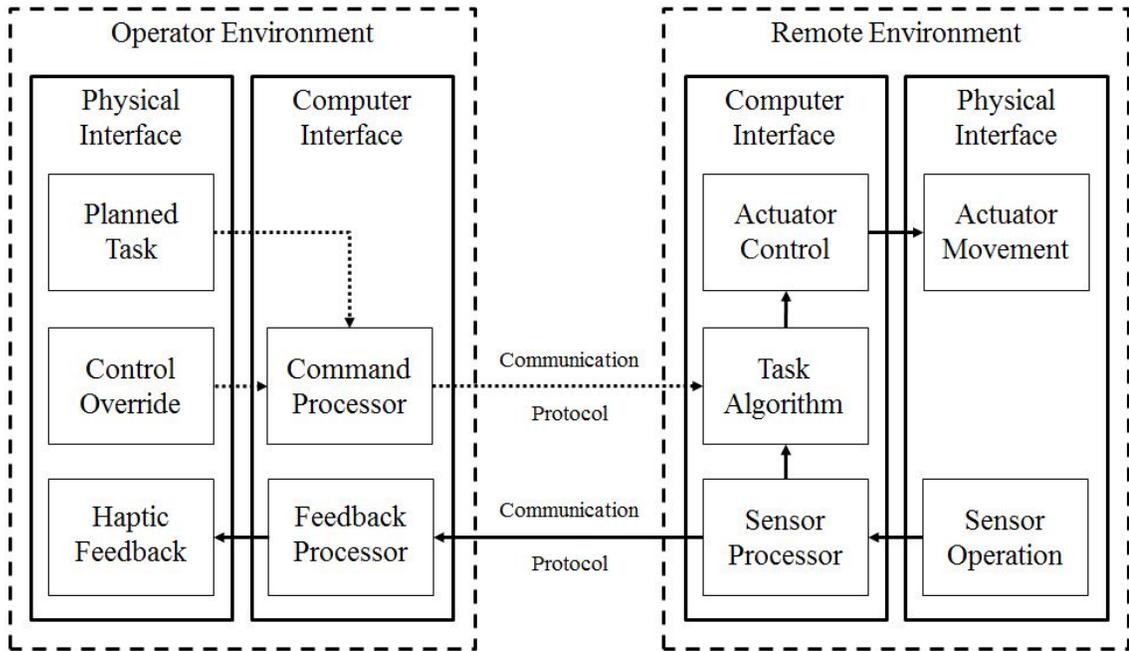


Figure 2-2. Operation of a supervisor-subordinate system

The next remote operation paradigm, partner-partner control, is unique in that the task is not performed in the remote environment, but in the operator's environment. The robot exists to support the operator's actions through perception and actions to aid in the performance of a task. Figure 2-3 shows that, while actuation and sensing do occur in the remote environment, they are used as feedback to assist in the performance of the operator's task.

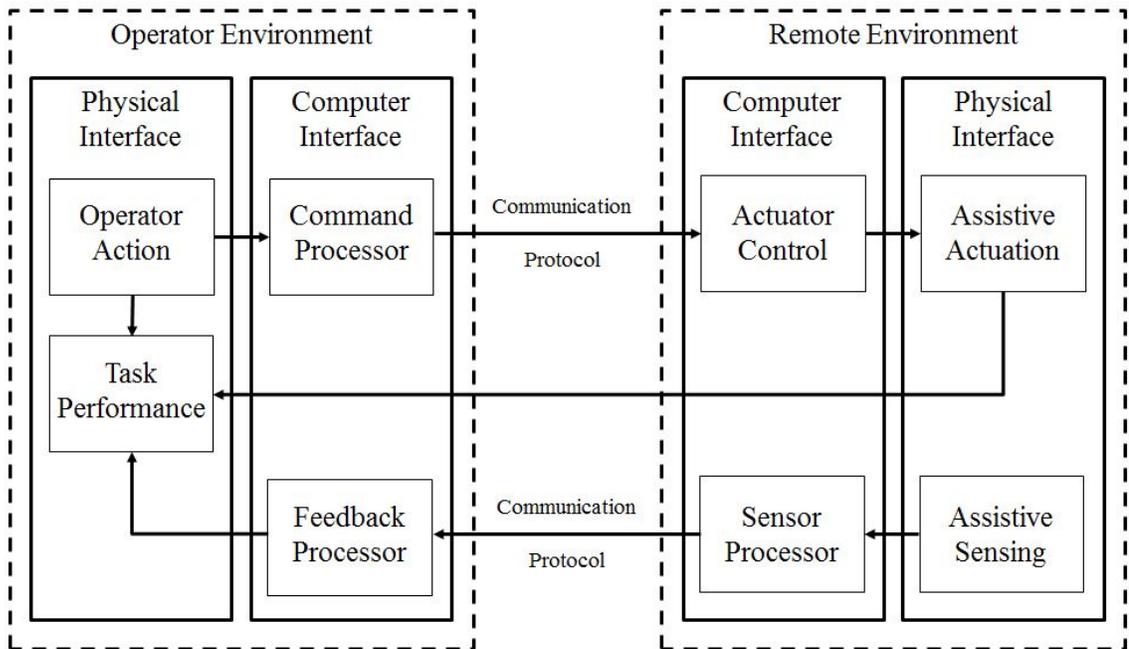


Figure 2-3. Operation of a partner-partner system

Teacher-learner control is a mix between a pure master-slave system and a supervisor-subordinate system where the operator controls the robot, but the robot learns from the operator's commands. This method is shown in Figure 2-4.

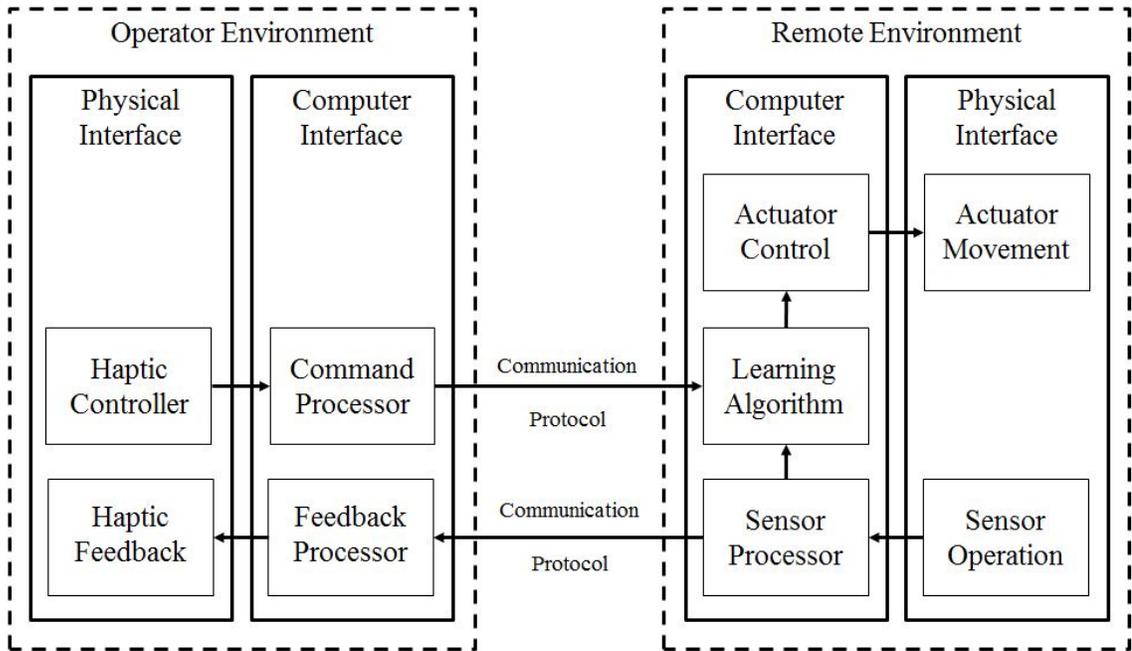


Figure 2-4. Operation of a teacher-learner system

Finally, fully autonomous remote control of robots, shown in Figure 2-5, are able to operate completely independent of operator interference and are able to guide themselves through their tasks without any assistance. Still, feedback may be sent back to an operator to monitor the robot.

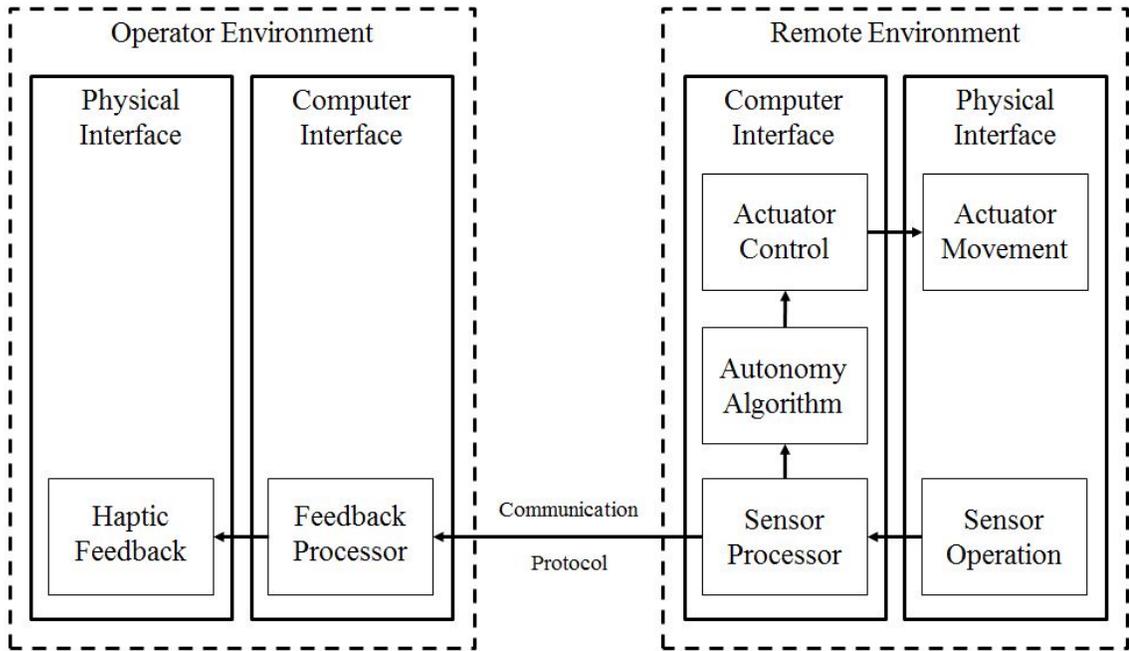


Figure 2-5. Operation of a fully autonomous system

This thesis will test both a pure master-slave system and a combination of a master-slave system and partner-partner system. While the majority of the teleoperation performed is pure master-slave, instances of remote vehicle intervention is a partner-partner system, with the remote vehicle assisting the human operator with additional local data, especially when such data is not directly available to the human operator. Additionally, it is important to note that, since this work focuses on vehicle teleoperation, the human operator is referred to as the human driver and the remote environment is referred to as the remote vehicle when specifically discussing the teleoperated vehicle.

2.1.2 Haptics

In teleoperation, haptics is essentially the physical, visual, or other feedback received by the operator to increase telepresence [2]. Haptic feedback comes in many forms, from visual

feedback from a camera to control the position of a robotic arm [19], to physical feedback from a vibration [2]. The necessity of feedback varies by devices, but some amount of feedback is necessary in all applications. For example, in the application of robotic hand, visual stimulation may be less important than physical stimulation when making minute movements, but it is very difficult to emulate the precise nerve feelings in a hand [2]. In this case some creative sensors must be utilized like vibration that changes strength as pressure increases or three dimensions of actuators on the fingertips that press against the user to mimic the pressure on the robot.

There are many mechanisms for feedback in remote driving. For example, at Carnegie Mellon University's Robotics Institute, Fong uses a combination of camera and range sensor data on a six-wheeled skid-steer vehicle to create force feedback on the operator's joystick [20]. To test the effectiveness of this haptic feedback, untrained operators were tasked with driving the robot through a maze in two conditions: with video feedback alone, and with video and force feedback on the joystick. They found that with the force feedback all operators were able to precisely navigate the maze, but only visual feedback resulted in numerous crashes and some were unable to accomplish the task.

Another study was performed by Toffin at the Renault Technical Center for Simulation, where a feedback torque was provided on a driving simulator steering wheel based solely on steering wheel angle [21]. A series of feedback models was provided, varying from a linear to a highly non-linear relationship between steering wheel angle and feedback torque. The plot showing the relationship between provided steering wheel angle and feedback torque is shown in Figure 2-6. The response from several human subjects showed that, while not fully realistic, feedback generated by a simple linear relationship improved the driving experience while driving with no feedback was very difficult. The full results from the experiment are shown in Table 2-1.

Table 2-1. RMS variations obtained in driving simulation experiment [21]

	Lateral acceleration (m/s ²)				Lateral deviation (m)					Steering wheel angle (°)					
	K1	K2	RMS (K2-K1)	K3	RMS (K3-K1)	K1	K2	RMS (K2-K1)	K3	RMS (K3-K1)	K1	K2	RMS (K2-K1)	K3	RMS (K3-K1)
S1	1.15	1.08	-0.07	1.35	0.20	0.45	0.39	-0.06	0.78	0.33	10.22	8.46	-1.76	11.8	-1.58
S2	1.44	1.46	0.02	1.14	-0.30	0.85	0.53	-0.32	0.89	0.04	22.68	11.87	-10.81	14.71	-7.97
S3	0.71	0.84	0.13	1.33	0.62	0.71	0.49	-0.22	0.91	0.20	18.04	9.88	-8.16	10.52	-7.52
S4	1.14	0.75	-0.39	0.36	-0.78	0.69	0.55	-0.14	0.77	0.08	14.9	9.17	-5.73	4.58	-10.32
S5	0.35	0.71	0.36	0.34	-0.01	0.41	0.4	-0.01	0.37	-0.04	5.41	11.02	5.61	5.04	-0.37

Note: Values below thresholds are marked in gray, and correspond to a good adaptation of drivers to changes in the steering feedback configuration.

Role of steering wheel feedback

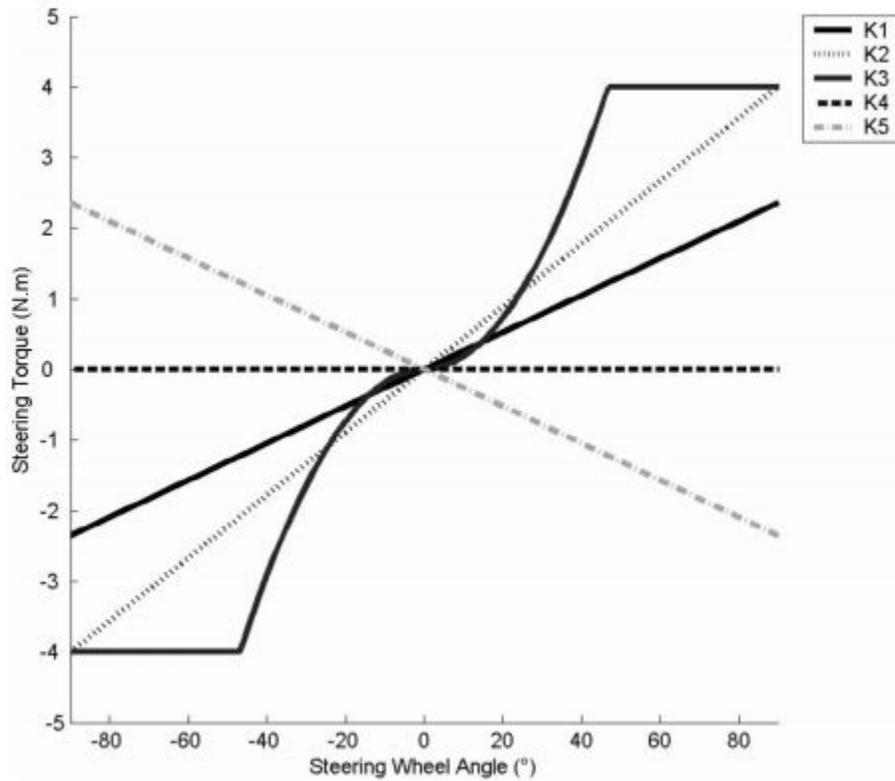


Figure 2-6. Plot of force feedback laws used by Toffin [21]

It is usually impossible to completely replicate all motions and forces acting on a remote robot, especially when considering all design aspects [22]. The choices of feedback types must be considered against each other in the full design process. One example of this creative design is performed by Shoal [23] who experimented with a way to detect pressure as the operator manipulated a robot without physical feedback. Similarly to a robotic hand, surgery tools were

being controlled remotely and required precise controls. Since physical feedback in this case can be extremely difficult to mimic, visual feedback was used in the form of a heads up display that showed the operator vital information such as time and tool pressure.

Human interactions with the environment are bi-directional. We simultaneously perceive and manipulate the environment at all times. However, manipulation and perception do not happen at the same rate. According to Tanner [22], manipulation occurs between 1-10Hz while tactile perception occurs around 1000Hz. While these are general guidelines, actual human response rate is dependent on its current function [24]. For example, while the human body may be able to detect vibrations at 50-400Hz, it can only detect compression on the body at ~10Hz. Brooks found that both proprioceptive feedback where receptors sense muscle contraction and tension, along with kinesthetic feedback where receptors sense angle and velocity of joint movement occur around 20-30Hz [24]. These are the functions primarily used in telerobotic response and vehicle steering haptic feedback. For full transparency within a remote vehicle context, it is necessary to meet these 30 Hz requirements to fully mimic the natural driving environment.

Along with possible slow performance, stability issues can arise during force feedback. Al-Mouhamed [25] performed an experiment using a master-slave robotic arm to create and maintain contact on a surface. The contacted material and the gain of the force feedback were altered to examine the performance of the operator. Al-Mouhamed found that stability is difficult to maintain during the initial point of contact and the point of release. The cause of this instability was a combination of the stiffness of the material, the feed forward gain, and the round trip delay which remained about constant at 183ms. The results from this work are shown in Figure 2-7 where (a), (c), and (e) are the force plot for contact with rubber, human, tissue, and a rigid object, respectively. Plots (b), (d), and (f) are the motion plots for the same materials. The instabilities at

the pre-contact and post-contact phases are evident, especially when in contact with a rigid material.

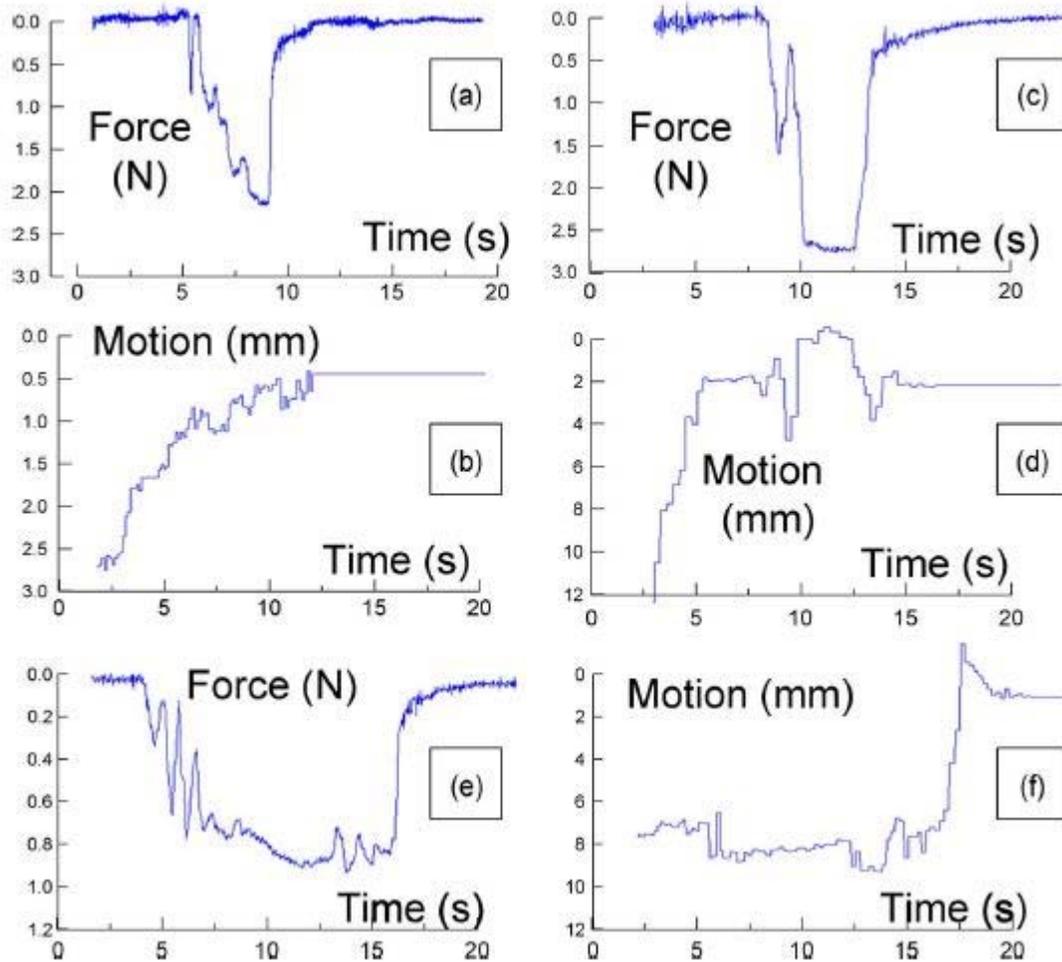


Figure 2-7. Force feedback responses with various materials [25]

2.1.3 Vehicle based teleoperation

Vehicle based teleoperation first became popular in the 1970's and currently covers many different domains including air, ground, and underwater [3]. Today, air vehicles, also known as Unmanned Air Vehicles (UAVs), are the most common type of teleoperated vehicles and are

used for many purposes from surveillance to combat [3]. One of the most common UAVs, the Predator drone, is shown in Figure 2-8.

Underwater vehicles, also known as Remotely Operated Vehicles (ROVs), are rapidly becoming the largest market for teleoperated vehicles, due to the difficulty of human survivability in deep water and the increasing interest in deep-water oil drilling. For example, Chouiten [26] performs augmented reality work on ROVs by integrating an existing map with an onboard camera on the vehicle. His ROV is shown in Figure 2-9. ROVs are usually tethered to another vehicle on the water surface and are used for tasks like surveillance, mapping, and oceanography [3].



Figure 2-8. Predator drone UAV [27]

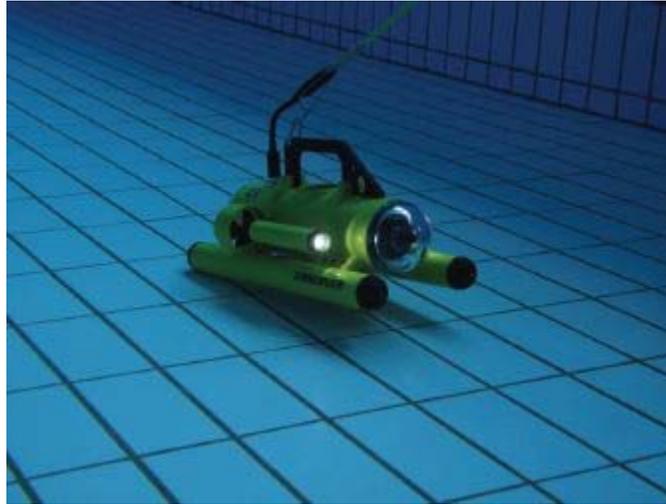


Figure 2-9. ROV used by Chouiten [26]

Ground vehicles, also known as Unmanned Ground Vehicles (UGVs) perform tasks from exploration to hazardous duties [3]. Exploration vehicles includes those like NASA's Lunokhod 1 moon rover, shown in Figure 2-10, while hazardous duty vehicles are ones that clean up hazardous chemicals or defuse bombs. Haptics has become one of the preliminary focuses of UGVs in recent years to increase the transparency of the system. In the work by Nguyen [28], multiple haptic feedback patterns are proposed based on a joystick steering system including position-position matching, position-velocity matching, and a mix of the two. This allows the operator to simultaneously send commands via the joystick and receive haptic feedback on the actual UGV position or velocity. According to the same researcher's earlier paper, haptic feedback could be achieved in the same way with a traditional steering wheel by performing position-position control on the angle of the steering wheel [29].

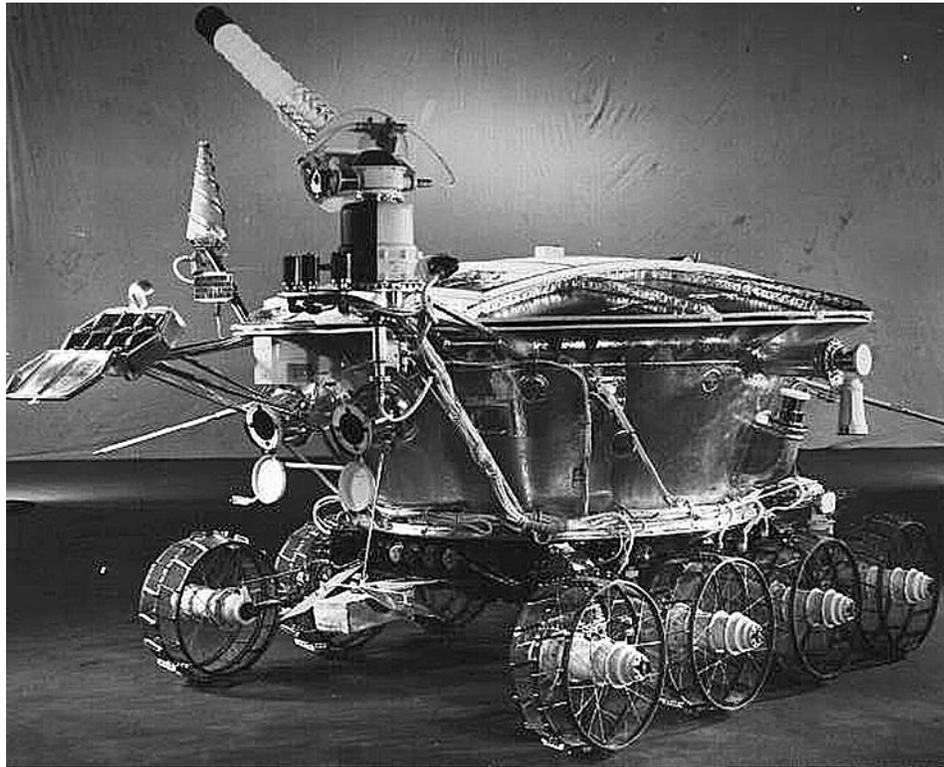


Figure 2-10. Lunokhod 1 moon rover [3]

2.1.4 Transparency in driving simulators

As the end goal of teleoperation is full transparency, or the complete immersion of the operator into the remote environment, the same is true for HIL simulators with the simulated environment. The difficulties in reaching full transparency are discussed extensively by Ersal [30]; specifically, transparency must minimize problems caused by delay, jitter, and packet loss. Delay is the time between data being sent and data being received, jitter is the variability in delay, and loss is the dropping of data packets. One way to mitigate these problems is to implement an observer model on the operator side, robot side, or both. The goal of this observer is to estimate states and control the robot in the event of data transmission problems [30]. Further, the coupling point between the HIL simulator and its hardware can be used as a design metric to minimize data

transmission issues [31]. Unfortunately, there are many cases where the hardware's physical size or location limits the ability to design around the coupling point.

There is a direct tradeoff between the transparency in the system and the expense of system implementation [32], meaning that full telepresence generally requires expensive equipment in most systems. This leads to a new design criteria in simulators and teleoperated systems: achieve the highest telepresence with minimum cost. In other words, the designer must consider the effect of telepresence on top of traditional design objectives like cost and safety when creating a new simulator.

2.1.5 Further areas of research

It is important to note other areas in which teleoperation has been implemented. One popular use of teleoperation, especially Internet-based teleoperation, is to connect students for real-time collaborative control of robots. Goldstain conducted experiments over a number of years to determine the best telerobotic interface with which to educate engineering students [33]. These experiments required Internet-connected students to move an object together using robotic arms, resulting in a set of procedures that best helps students understand the operation of robots. In a separate study, Tanaka linked young school children in Japan and the United States to cooperatively interact with robotic arms, finding that a delay up to 500ms exists in the communication [34]. This delay dramatically reduced the effectiveness of the program, but no expansion has yet been made on this experiment.

One unique application of teleoperation is the goal of creating telepresence in an art museum [35]. While this work is seemingly unscientific on the surface, it discusses many ideas that are applicable to creating a high level of transparency in any teleoperation situation. Specifically, the project utilizes a combination of virtual reality to give users an ability to explore

the museum and interact with others and direct camera view to get a full realistic view of the paintings via high definition camera on a teleoperated robot. This combination of realities allows users to get a fuller experience of exploring the museum and immersion into the environment than to implement a purely virtual or a purely camera based reality.

2.2 Latency issues and testing procedures in telerobotics

2.2.1 Internet based teleoperation

As stated in Section 1.1 there are many ways to communicate with a teleoperated robot, for example, via radio or Bluetooth communication. This thesis will rely primarily on Internet communication, so a brief review will be performed focusing on the tradeoffs between each communication medium for teleoperation.

Internet communication has been used in both wired and wireless cases of teleoperation. One work by Marin uses a wired Ethernet connection to control a robotic arm [36]. Using Java, Java3D, and CORBA, commands are sent over this wired connection and a camera image is sent back as feedback to simulate the robotic arm in both virtual and augmented reality. Other researchers like Moutaouakkil [37] rely on a wireless Internet connection to simply send commands to a Lego Mindstorms robot. While Moutaouakkil [37] may be a short-distance example of wireless Internet communication, similar wireless connectivity can be done over any distance. In one piece of research by King, et al., six Universities world-wide participated in a 24 hour event to test and demonstrate teleoperated surgical tools [38]. Unfortunately, exact bandwidth values are not provided in any of these works.

Vehicle teleoperation has been performed using wireless Internet based platforms. Grange [39] developed a platform called WebDriver to allow vehicle teleoperation minimizing

bandwidth usage, providing sensor fusion displays, and optimizing human-computer interfacing. The same research group also developed PdaDriver which uses a Personal Digital Assistant (PDA) to teleoperate a vehicle in low-bandwidth/high-latency communication links, although specific values for bandwidth and latency were not presented [20]. All of these works emphasize a need to develop low-bandwidth systems due to cost restrictions; however none provide numerical benchmarks for achieving low-bandwidth.

Since many sources do not reference exact bandwidth values achieved during testing, alternative sources were consulted to determine standard data transmission rates for a variety of transmission mediums. Modern wireless interfaces to the Internet have a wide range of bandwidths depending on the protocol, but the most common currently used protocol at present is 802.11n which has a maximum transmission rate of 600 Mbps [40]. Other wireless protocols are also used, including Bluetooth which has a maximum data transmission rate of 720 kbps [41] and dedicated short-range communication (DSRC) which has a 500 kbps rate when operating within 1500ft, and a 27 kbps rate when operating between 1500-3000ft [42].

Latency is as equally important as bandwidth for system performance. Interestingly, bandwidth and latency have an inverse relationship in quality for the Commercial Off-the-Shelf (COTS) systems investigated in this study. For example, comparing 802.11n to both Bluetooth and DSRC communication systems, 802.11n has the best bandwidth yet has the worst latency at about 40ms [43]. Bluetooth has slightly better latency at about 25ms for the first few packets sent, but it improves slightly after this initial transmission delay [44]. DSRC boasts the lowest latency times at only about 2ms, but it has the worst bandwidth [45]. Figure 2-11 shows a comparison of the latency and bandwidth for these three communication protocols.

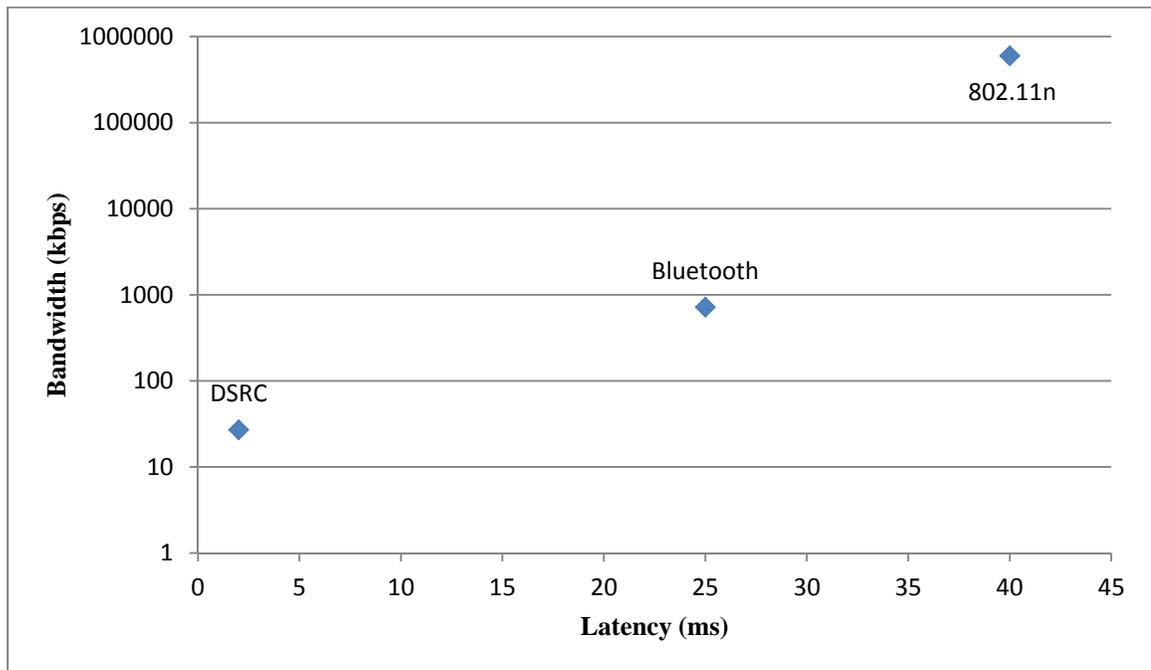


Figure 2-11. Comparison of latency and bandwidth for three communication protocols

2.2.2 Strategies for controlling system latency

While Internet communication can greatly increase the effective distance of teleoperation, it can also greatly increase time delays in the system. One area of research focuses on the reduction of these time delays. Ploplys et al. [46] performed research to develop real-time control systems over wireless networks. In this work, a pendulum was controlled by a linear quadratic regulator (LQR) state feedback method for stabilization. The LQR controller was implemented without taking the effect of transmission delay into account. Data was transmitted over an 802.11b line and round-trip transmission time was calculated, finding that the round trip transmission time for 84 bytes of data was between 3.3 and 3.8 ms, as shown in Figure 2-12. Ultimately, this work found that the LQR controller stabilized the pendulum for various sampling rates as long as under 22% of packets were lost. Full results are shown in Figure 2-13.

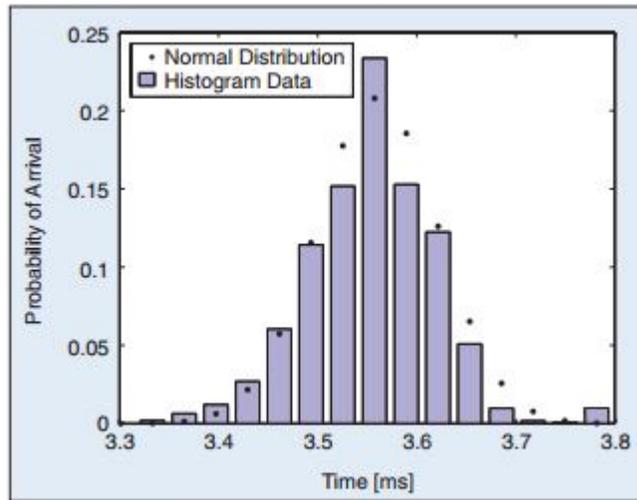


Figure 2-12. Round trip response times for 84 byte data packets [46]

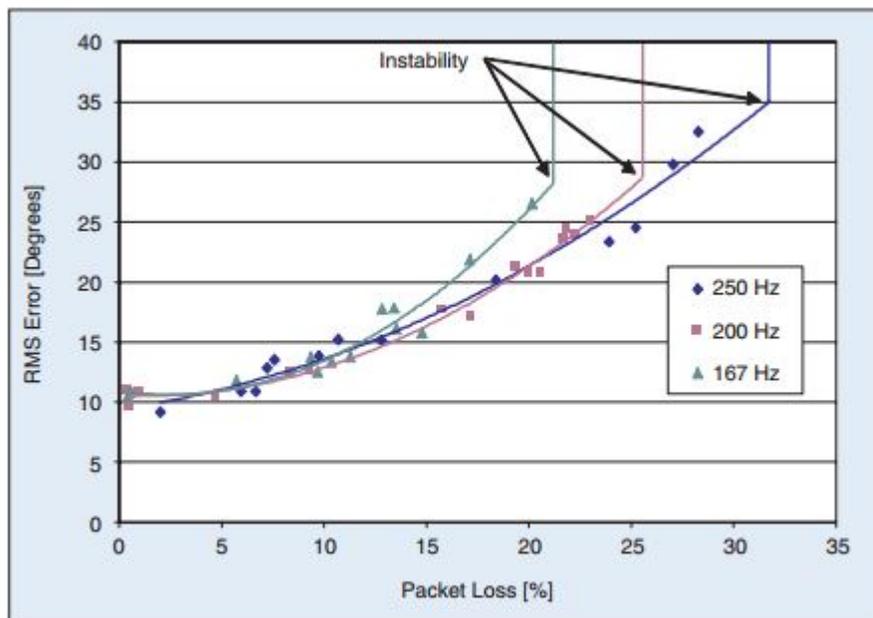


Figure 2-13. RMS error of pendulum position for various sampling rates [46]

Later research by the some of the same authors uses linear matrix inequality (LMI) for stability and performance analysis and a stochastic two-state Markov model for control in a packet-based environment [47] [48]. The Markov chain is a simple tool that is commonly used for analyzing the effectiveness of packet-based systems. As shown in Figure 2-14, the two states,

open loop and closed loop, have some probability (p) that a packet will be received and the system will move onto the other states and one minus that probability ($1-p$) that a packet will be dropped, in which case it remains on its own state. Packets are considered dropped if transmission is lost, the data is corrupted, or the delay exceeds sampling time.

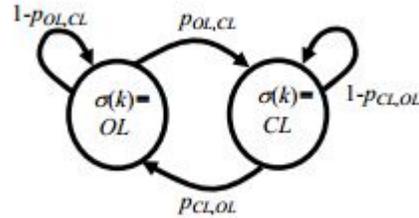


Figure 2-14. Two-state Markov chain between open and closed loop system [48]

Three strategies are presented for controlling the system during drops predicted by the Markov model. First, the control effort is simply set to zero. This method is typically undesirable, as is better suited as an additional condition for the second strategy, Zero Order Hold (ZOH). ZOH utilizes the control law shown in Equation 2-1 to choose an input to the system. If the packet is delivered, then the controller proceeds as normal. If up to N packets are dropped, then the controller outputs the last command. If more than N packets are dropped, then controller outputs zero.

$$u(k) = \begin{cases} Ke(k-1) & \text{if success (CL)} \\ u(k-1) & \text{if } \leq N \text{ consecutive losses (OL)} \\ 0 & \text{if } > N \text{ consecutive losses (OL)} \end{cases} \quad \text{Equation 2-1}$$

The third strategy is estimation, where the controller continues to estimate the states of the model in the event of dropped packets. The input is selected based on Equation 2-2, which is similar to Equation 2-1 only with a different second condition that calculates a controller based on the state of the system instead of simply outputting the past value. This condition is shown in Equation 2-3.

$$u(k) = \begin{cases} Ke(k-1) & \text{if success (CL)} \\ u^*(k) & \text{if } \leq N \text{ consecutive losses (OL)} \\ 0 & \text{if } > N \text{ consecutive losses (OL)} \end{cases} \quad \text{Equation 2-2}$$

$$u^*(k) = \begin{cases} Kr(k-1) - K[A \ BK] \begin{bmatrix} x(k-2) \\ e(k-3) \end{bmatrix} & \text{if 1 loss} \\ Kr(k-1) - BKr(k-3) - K[A \ BK] \begin{bmatrix} A & BK \\ -I & 0 \end{bmatrix} \begin{bmatrix} x(k-2) \\ e(k-3) \end{bmatrix} & \text{if 2 losses} \\ \vdots & \vdots \\ K \left[r(k-1) - \sum_{j=-1}^{N-3} A^{N-3-j} BKr(k-N+j) \right] - K[A \ BK] \begin{bmatrix} A & BK \\ -I & 0 \end{bmatrix}^{N-1} \begin{bmatrix} x(k-2) \\ e(k-3) \end{bmatrix} & \text{if } N \text{ losses} \end{cases} \quad \text{Equation 2-3}$$

Instead of modeling the system to control it during a large time delays, some researchers examine how time can be replaced as a variable. Elhajj [49] discusses the idea of using an event-based controller instead of a time-based one. With time-based controllers the dynamics of the system are modeled by differential equations using the time variable as a reference. With event-based controllers, on the other hand, the dynamics of the system are modeled based on a “motion reference” or “action reference.” This controller, as shown in Figure 2-15, is meant to reduce instability, loss, and de-synchronization caused by time delays. In the figure the “planner” is the operator sending commands to the remote (“robot”) environment. The work by Elhajj [49] uses event-based control to send super-media, or a massed collection of data, over the Internet in a more stable and less delayed manner. Various teleoperation setups are used that show effectiveness in transmitting velocity commands from the operator and video, temperature, and force feedback from the robot arm. One set of results that demonstrate the difference in time-based response (left) and event-based response (right) is shown in Figure 2-16.

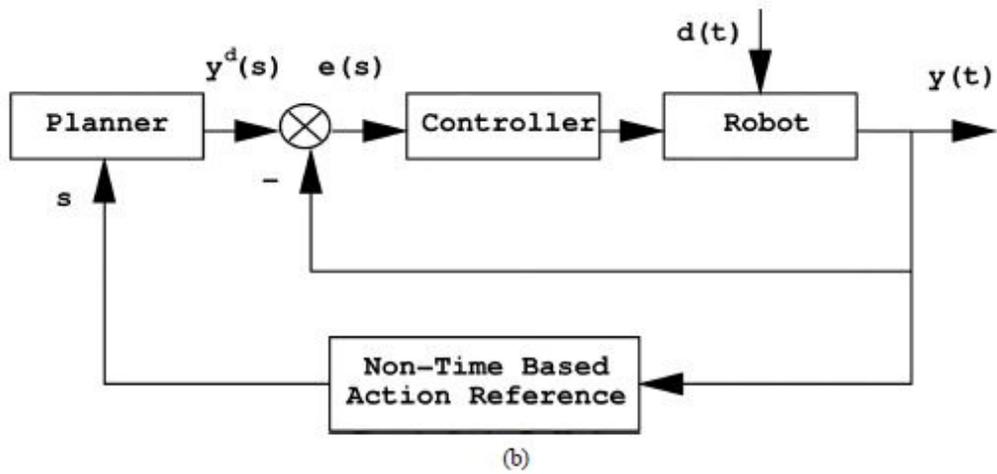
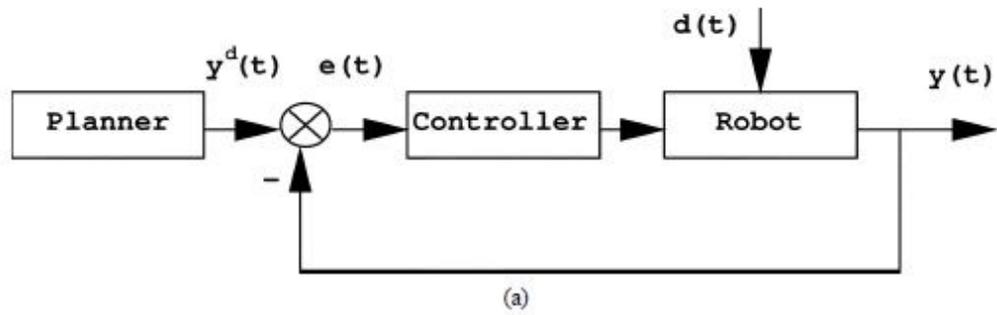


Figure 2-15. Traditional time-based control (a) and event-based control (b) [49]

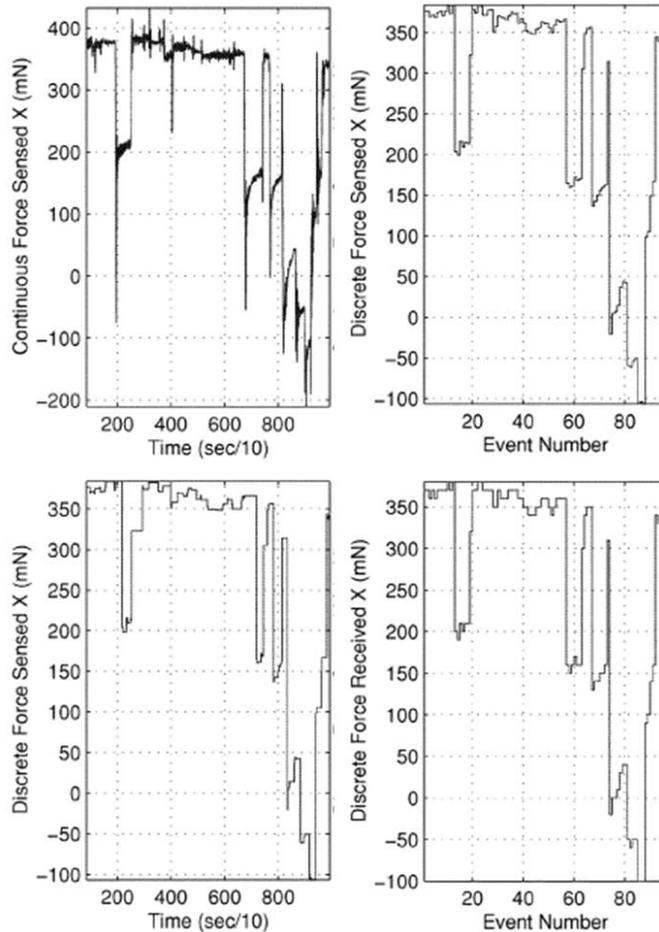


Figure 2-16. Comparison of time-based response (left) to event-based response (right) [49]

2.2.3 Common testing procedures

It is desirable to perform tests of telerobotic systems that produce results that are comparable to each other or to current standards. For this reason a review of other testing procedures in teleoperation research is presented here. The vast majority of works simply plot the time of the command from the operator vs. the response time of the remote robot [32] [50] [51]. One example of this is shown in Figure 2-17. It is important to note that in this type of testing a timestamp must be sent with every message to accurately represent the delays.

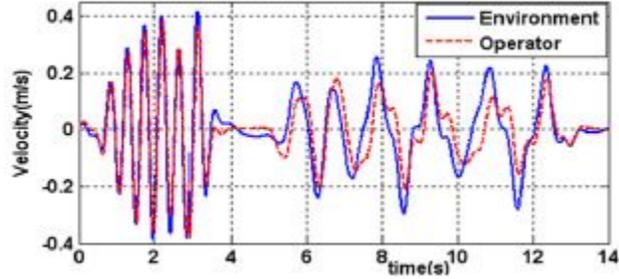


Figure 2-17. Example command vs. time plot for operator and robot [50]

While there is not a consistent movement pattern that is imposed or repeated between any of the works listed above, they do discuss specific movements unique to each experiment. For example, the brief work by Wen [51] discusses the control of a hydraulic robot arm by sending either a displacement or force command. In this work, the motion recorded is the force and displacement of the robotic arm as it is making and releasing contact with various materials.

On top of a comparison of master and slave response, some works include frequency-domain responses in their analyses. In work by Tanner [22], the position of a master arm is controlled by a human user while a slave arm provides force feedback to the through a time-delayed medium. The system is augmented with various feedback filters to improve arm response at higher frequencies. These data are presented in magnitude bode plots, as shown in Figure 2-18. This method of testing data is useful because frequency responses are universal to all dynamic systems.

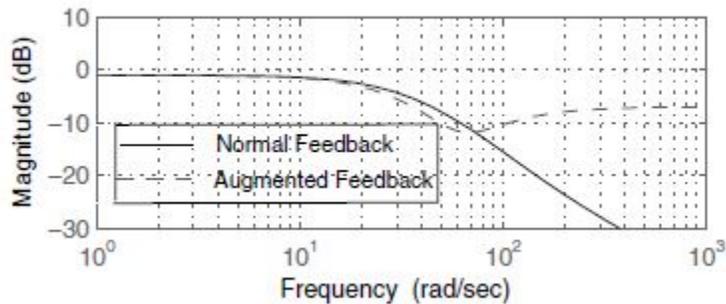


Figure 2-18. Bode plot showing improved high frequency response [22]

Some researchers base success in teleoperation by achieving high data transfer rates and low delays. In the work by King [38], six Universities across the world ran a 24 hour master-slave surgical tool trial to examine effectiveness of teleoperation via internet connection and presented the average data transfer rates. While the desired rate as mentioned in Section 2.1.2 is 1000Hz, these researchers found they averaged only around 10Hz. In the work by Harder [52] data on video streams and system control is collected including characterization of one-directional delays in the communication (in milliseconds), jitter (in milliseconds), data transfer rate (in MBPS), and bit error ratio. These data are shown in Table 2–2. Controlling the system through delays is a primary focus of this work over reduction of delays. For this reason, these testing procedures will not be utilized in this work.

Table 2–2. System performance data [52]

Parameter	Measurement #1	Measurement #2	Measurement #3
Video Data Stream			
$t_{L,V}$	51.7 ms	59.2 ms	148 ms
PJ_V	5.68 ms	11.2 ms	13.1 ms
DR _V	3.75 Mbps	3.76 Mbps	3.77 Mbps
Lost Pack _V	0.024 %	0.104 %	0.020 %
Spacecraft Control Data Streams			
t_{SC}	15.4 ms	497 ms	847 ms
DR _{UplinkSC}	31.8 kbps	31.2 kbps	32.8 kbps
DR _{DownlinkSC}	0.22 Mbps	0.22 Mbps	0.23 Mbps

Some works simply list the total time to complete a task in order for comparison against other test subjects [18] [53]. Goldstain [33] uses this strategy to test the effectiveness of telerobotics as a teaching tool by measuring the number of steps each test subject commands and the number of errors each subject commits. Since it is very difficult to measure system performance against dynamically unrelated teleoperation systems using these methods, they will also not be implemented.

In terms of vehicle teleoperation experiments, Halme [32] tests the teleoperation of an ATV using a set track in a corridor or slalom, performing obstacle avoidance in open terrain,

loading a vehicle, stopping precisely from high speeds, and a combination of this in off-road driving. Along with graphical actuator response times, further testing is done by having five subjects drive the vehicle and answer subjective questions involving various aspects from the ease of driving to ergonomic drawbacks. Similar testing is also performed by Nguyen [28] where ten subjects are instructed to drive a three wheel vehicle using a variety of joysticks with different haptic feedback mechanisms on the same figure-eight type course. Time of course completion for each feedback mechanism was plotted and the subjects were provided subjective analysis on the response for various haptic feedback methods.

This work is unique in that, instead of detecting latency, the system is altered in various ways to detect how the driver responds to latency issues. More specifically, one set of experiments will require a human operator to teleoperate a vehicle at constant speeds with reducing preview time to determine the relationship of the driver's phase margin and preview time. The other set of experiments requires the human operator to teleoperate a vehicle at constant speeds while increasing delays are added to the system. These tests will provide an understanding human sensitivity to both delay and risk when teleoperating a vehicle.

CHAPTER 3

SOFTWARE ENVIRONMENT

The piece of software used in this project is an open source platform called Robot Operating System (ROS). As this software is used to both control the test vehicle and manage the communication for teleoperation, a basic understanding is necessary. This chapter presents an overview of ROS and an explanation of why it was chosen for this project.

3.1 Overview of ROS

ROS is a Linux-based open-source software platform that provides libraries and tools to help designers create robot applications [54]. It was originally developed in 2007 by the Stanford Artificial Intelligence Laboratory as part of the Stanford AI Robot STAIR project, but since 2008 it has been primarily developed by Willow Garage, a robotics research institute with more than twenty institutions collaborating on robotics development. ROS is designed to provide an “operating system-like functionality” for robot software development. It is compatible for programming with both Python and C++.

At its most basic, ROS operates as a cluster of *nodes* operating individually to subscribe to, process, and publish data. Messages that are published and subscribed to are called *topics*. When one node publishes a topic it is typically available for all other nodes to subscribe to as needed. Figure 3-1 shows a basic example of how ROS nodes function. In this example a joystick is being used to teleoperate a robotic arm with some sensor feedback. The peripheral input processing node reads the data from the joystick, processes it, then publishes information. The sensor processing node does the same for the sensors on the robotic arm. The robot controller

node then subscribes to both the topic published by the peripheral input processing node and the sensor processing node, processes that data into a command for the robotic arm, and publishes that command.

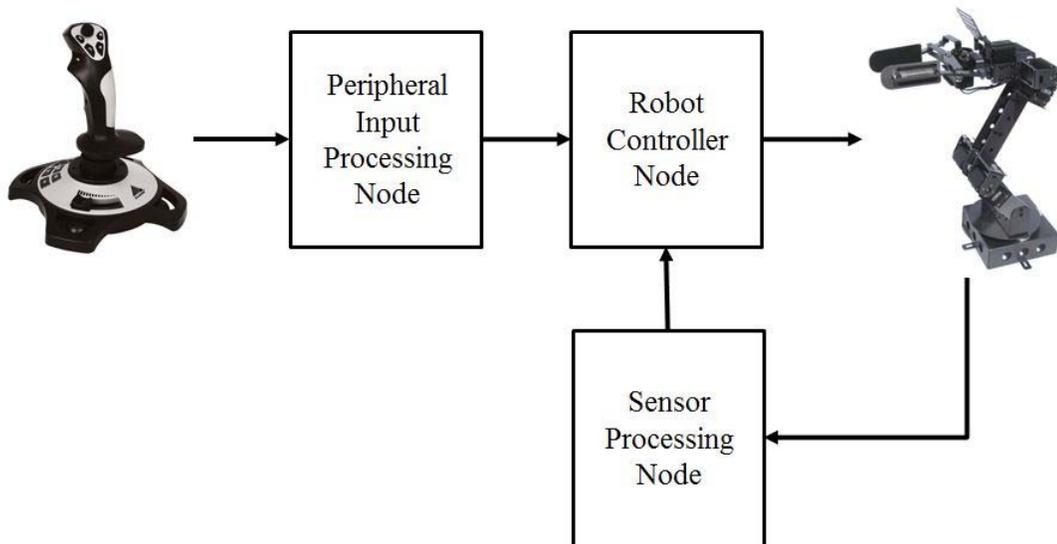


Figure 3-1. Basic ROS node diagram

By using this basic subscribe and publish structure virtually any hardware can be communicated with and controlled through ROS. This “middleware” aspect of ROS provides a highly modular design, yet can also be challenging to learn.

3.2 Justification of use

The decision to use ROS above other simpler programs like MATLAB’s xPC was based on the goal of having a highly customizable system. ROS does have a much steeper initial learning curve than xPC, but once that initial curve is overcome, any subsequent system expansion is comparatively simple. The ROS Wiki page provides a number of predesigned

packages called *stacks* that can help implement functionalities like perception, object, identification, motion control, and path planning, among many others [54]. More importantly, however, since ROS is an open-source platform there are a number of user-designed stacks that are premade for a large assortment of hardware devices. In contrast, a commercial product only gives access to what has been released by the company. In situations where the commercial software does not work, one is left either trying to reverse-engineer the software capability, or work through a sometimes lengthy back-forth process with product support experts.

One of the greatest benefits of the ROS platform is the ability to easily communicate across many devices via TCP/IP connection, allowing different CPUs to control different functions. Some specific benefits of this include the ability to:

- display large numbers of camera views across multiple CPUs
- separate CPUs for specific purposes, for example a dedicated DAQ CPU
- read inputs from multiple drivers simultaneously, including algorithm representations of “virtual” drivers
- interface with real vehicles

The ability of ROS to communicate across multiple platforms easily makes it ideal for teleoperation. ROS is able to process and collect data in a remote environment and publish that data back to the human operator. By manipulating the data in the remote environment, it is possible to artificially control the feedback provided to the human driver.

3.3 ROS to ROS communication latency

This thesis focuses on communication between ROS-enabled CPUs for teleoperation and thus latency anywhere in the system must be understood. To investigate sources of latency, it is important to know first what effect the ROS middleware itself has on that overall latency. To determine latency within the ROS software-to-hardware interface, two loopback tests were performed to determine the round-trip delay: one with a wired connection through an Ethernet switch, and one with a wireless connection through a wireless router.

An intranet was set up between an ASUS U47A laptop with an Intel Core i5 3.3GHz processor, 6GB ram, and a 750GB hard drive, and an ASUS CM6850 desktop with an Intel Core i7 3.4GHz processor, 8GB ram, and a 1TB hard drive. An N600 Netgear dual band router/Ethernet switch with 600Mbps wireless transmission and 1Gbps wired transmission was used for computer to computer communication. These were the only computers on the network during the test.

To perform the test, a simple ROS node was written that publishes a 1Hz square wave. A second ROS node was also written that subscribes to this topic and republishes the same data as a new topic. A universal time is kept between the two CPUs and also published. The data from these topics are then compared and the time between sending and receiving the square wave is calculated. The numerical results for these tests are shown in Table 3–1 while the plots are shown in Figure 3-2 and Figure 3-3.

Table 3–1. Results of ROS to ROS latency tests

	Wired	Wireless
Average Latency (ms)	1.0	1.0
Max Latency (ms)	1.2	1.5
Min Latency (ms)	0.83	0.010
Standard Deviation (ms)	0.041	0.11

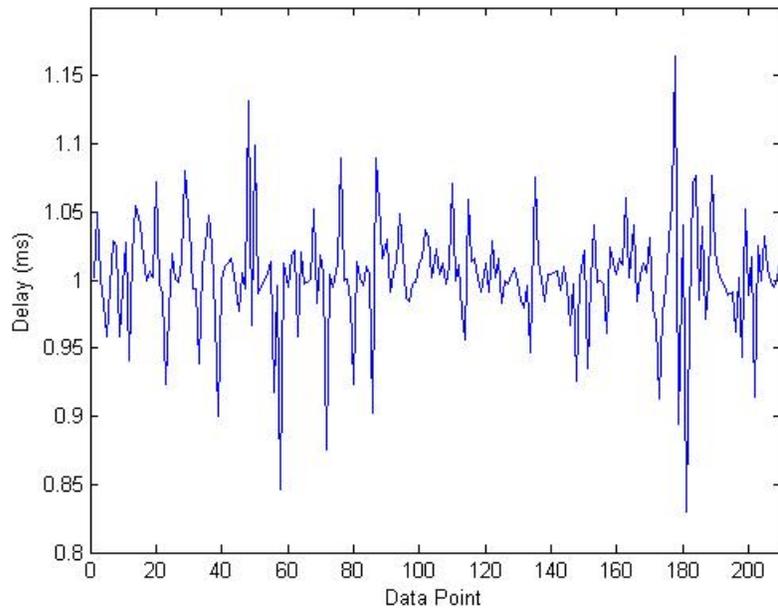


Figure 3-2. Latency for a wired ROS to ROS connection

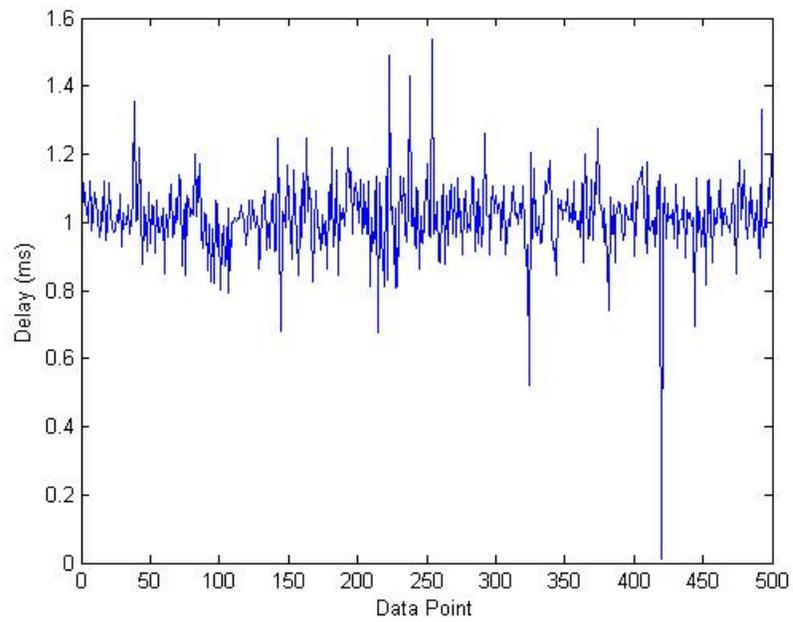


Figure 3-3. Latency for a wireless ROS to ROS connection

CHAPTER 4

TESTING PROCEDURE AND PLATFORM

As discussed in Section 2.2.3, a variety of tests have been designed to determine how various forms of feedback are used by human drivers during vehicle teleoperation. This section describes how tests were designed and implemented. Additionally, the vehicle used in testing and the model used to simulate this vehicle are both described.

4.1 Test vehicle

4.1.1 Physical vehicle description

The vehicle used for testing is a 1989 GMC 2500 referred to by this research group as “Big Red.” This vehicle was used for testing because it was instrumented to collect a variety of data from steering angle to LiDAR scans. Additionally, the parameters of this vehicle have previously been thoroughly tested, as documented in Lapapong’s prior work [55]. These parameters are shown in Table 4–1. The test vehicle itself is shown in Figure 4-1.

Table 4–1. Vehicle parameters for test vehicle

Symbol	Value	Unit
m	2579	kg
b	1.964	m
a	1.39	m
I_{zz}	5411	kg-m ₂
C_{af}	-75700	N/rad
C_{ar}	-83700	N/rad



Figure 4-1. 1989 GMC 2500 “Big Red”

4.1.2 Hardware layout

While the truck is equipped to gather a large variety of data, the minimum amount of data is collected for each test to conserve bandwidth. The data required for this test is differential GPS (DGPS) data for vehicle motion, camera data, and both commanded and response steering data.

The DGPS system used in this work is a Novatel SPAN RTK GPS, shown in Figure 4-2, with a 2cm accuracy that is considered precise enough to be used as ground truth. DGPS data was collected at 50Hz. The camera used to collect vision data is the PointGrey Research Firefly MV USB 2.0 monochrome camera with VGA resolution. This camera, shown in Figure 4-3, was used because it has low pixel-level noise and produces a good video quality at a low price of \$35. The decision to use a monochrome camera was intentional: the goal was to limit the information passed to the driver for the test in a manner that emulates the information that would be available

to a low-cost automated driving system. While the effect is likely small, the colors provided by the ambient environment are assumed to not affect the driver's response.



Figure 4-2. Novatel SPAN RTK GPS



Figure 4-3. PointGrey Research Firefly MV USB 2.0 camera

The steering command is generated by a Logitech G25 racing wheel, shown in Figure 4-4. This device was chosen because it is capable of emulating the motion of the truck's steering wheel, but can also be immediately integrated into ROS with ROS's included joystick drivers. While the device includes foot pedals and transmission control, these were not used in this work for safety reasons. These steering commands are sent to an Aerotech rotary stage attached to the steering column that acts as a servomotor and controls the steering angle. The assembly of this steering motor on the steering column is shown in Figure 4-5.



Figure 4-4. Logitech G25 racing wheel

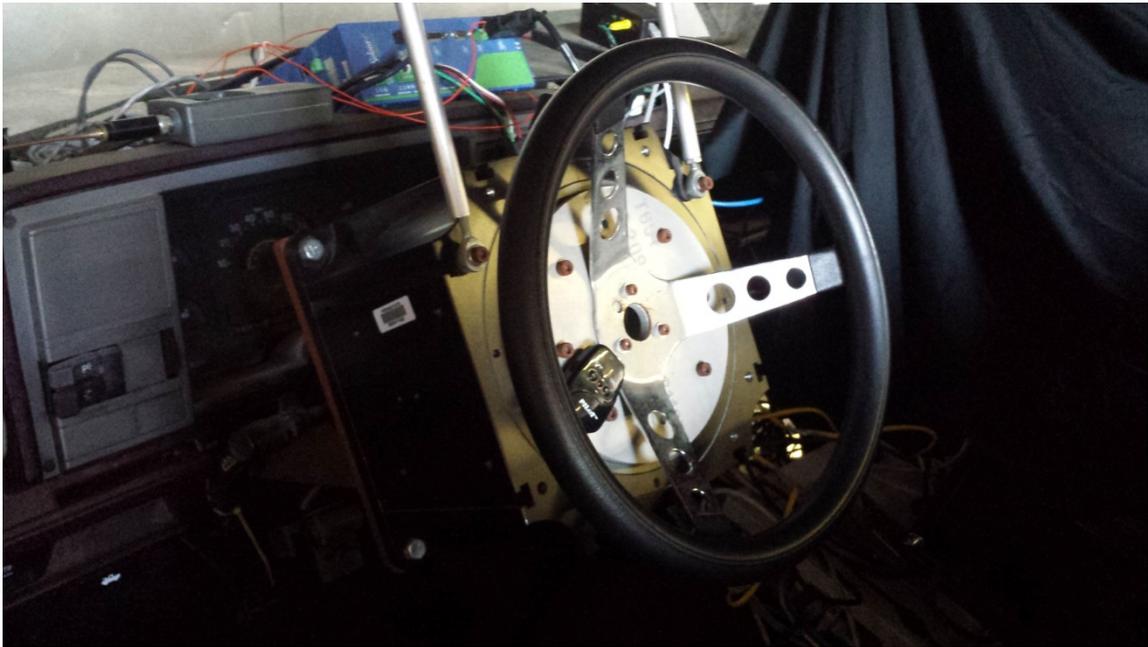


Figure 4-5. Aerotech rotary stage mounted on the steering column

External hardware can communicate with a ROS-enabled CPU in a number of ways. In this work, the Novatel GPS and Aerotech steering motor communicate via TCP/IP connection while the Logitech racing wheel and PointGrey camera communicate via USB. Since this project involves the teleoperation of a vehicle, two connected ROS-enabled CPUs are required. One computer displays the camera data and publishes steering commands from the human driver. The other records and publishes the camera data. The Novatel GPS and steering motor communicate directly with any of the CPUs via TCP/IP. The ROS-enabled CPUs also communicate via TCP/IP connection. All TCP/IP connections are routed through a Netgear N600 wireless dual band router capable of transferring up to 600Mbps wirelessly and 1Gbps wired. The hardware diagram for this system is shown in Figure 4-6.

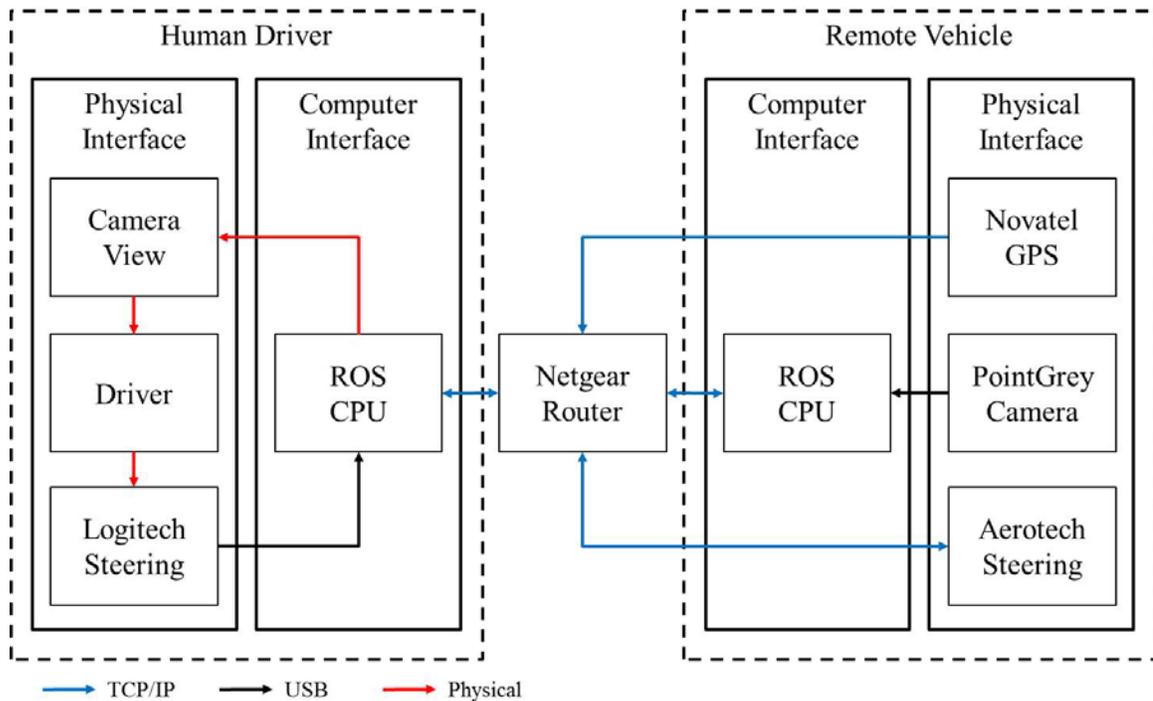


Figure 4-6. Hardware diagram for the test vehicle

4.1.3 Hardware latency and bandwidth analysis

As discussed in Section 3.3, it is necessary to know the latency in the transmission of data between the hardware to have a full understanding of the latency in the overall system. As shown in Figure 4-6, all data transmission between the hardware and ROS, with the exception of the steering motor, is single direction so a loopback test cannot be performed. For the Novatel GPS and Logitech steering wheel, the only way to test the interface timing is to read the frequency in which data is processed using a function included in ROS. The resulting transmission rates are shown in Table 4-2. Since the PointGrey camera is passed from one ROS-enabled CPU to another, the difference in clock time between the two CPUs can be used to determine latency. The latency for the PointGrey camera is shown in Table 4-3.

Table 4–2. Data refresh rates for test vehicle hardware

Hardware	Rate (Hz)	Latency (ms)
Novatel GPS	78.42	12.75
Logitech Steering	10.23	97.75
PointGrey Camera	15.03	66.53

Table 4–3. Wired and wireless latency for PointGrey camera

Connection	Mean Latency (ms)	Min Latency (ms)	Max Latency (ms)
Wired	100	30	155
Wireless	125	30	255

It is also important to know the total bandwidth of the system. ROS has the capability to record the bandwidth of all topics being published. These bandwidths are shown in Table 4–4.

Table 4–4. Bandwidth of vehicle hardware

Data	Bandwidth (KB/s)
GPS/IMU	7.71
Logitech Steering	6.23
PointGrey Camera	15,690
Steering Motor	0.0652

Because the steering motor has full feedback, the delay in motor response is measured by comparing the commanded motor position to actual position. The response was measured with the truck parked on the ground, but with the engine running to provide the use of power steering. The plot in Figure 4-7 shows the response of both a 90° commanded angle and a 180° commanded angle to determine if the motor is rate limited. Because the slope of the feedback data is the same for twice the input, the motor is clearly rate limited. The slope of the feedback lines indicates that the rate limit is about 725deg/s. Additionally, the motor response itself is non-linear partially due to what Brown [56] refers to as “bushing compliance.” The linkages between the steering rack and the wheels are not completely stiff, resulting in a slight lag when changing the steering direction and a non-linear relationship between steering command and steering angle.

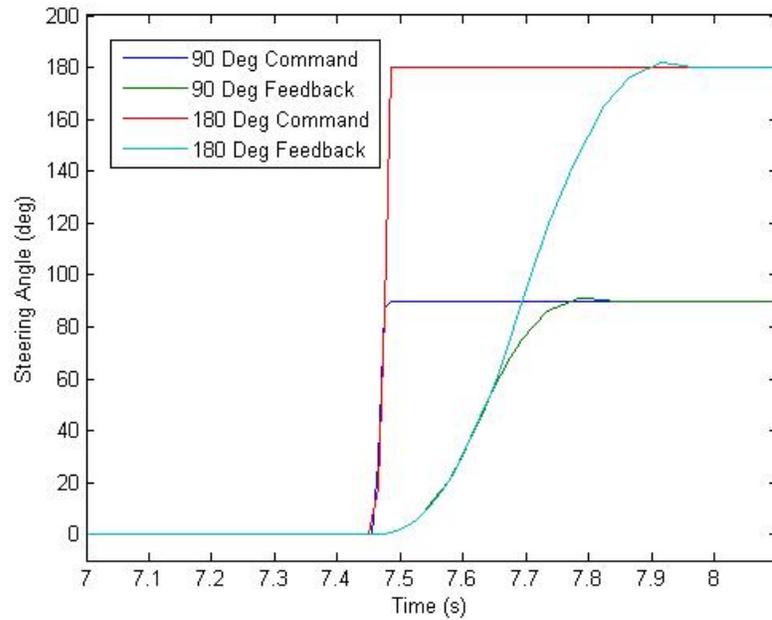


Figure 4-7. Steering motor response for a commanded 90° and 180° wheel angle

This plot aids in the generation of a dynamic model for the motor. Since the motor appears to be non-linear with the rate limit, a common 2nd order approximation will not fully capture the motion. However, it is useful as a baseline from which to create a more accurate model. To create the 2nd order model, Equation 4-1 and Equation 4-2, along with the damping ratio and rise time recorded from Figure 4-7 were used. The overshoot in the motor response is very minimal, indicating that the system is slightly less than critically damped. For this reason, a damping ratio of 0.75 was chosen. Since the system is rate limited the rise time will differ for the 90° and 180° commands. The rise time of the 90° turn was used because it has a lesser effect from the rate limit. This rise time is about 0.15s, leading to a natural frequency of 12s⁻¹.

$$t_r = \frac{1.8}{\omega_n} \quad \text{Equation 4-1}$$

$$\frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \text{Equation 4-2}$$

Due to the non-linearity, this model did not match well with the experimental data. From trial and error, the natural frequency was increased slightly to $16.5s^{-1}$ and the damping ratio was decreased to 0.65. Additionally, a 1st order transfer function was added to create the overall 3rd order effect of the response, which is likely due to an integrator-type control loop within the Aerotech motor controller. A delay block and two rate limit blocks were added to replicate the non-linear and delayed timing effects. The Simulink diagram for the steering motor model is shown in Figure 4-8.

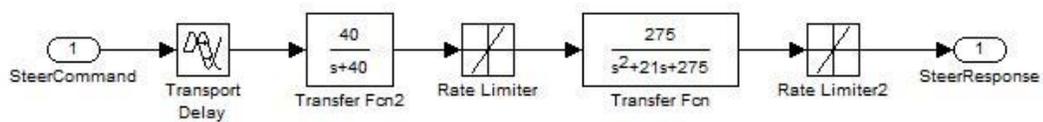


Figure 4-8. Simulink diagram for the steering motor

For analysis later in this work, it is necessary to form a transfer function for the overall steering motor model. The rate limiters are ignored in this estimated transfer function and the 0.02s transport delay is converted to a transfer function through a 1st order Pade approximation, resulting in a 4th order system. The final transfer function for the steering motor is shown in Equation 4-3. The eigenvalues of this transfer function are shown in Table 4-5.

$$\frac{SteerOut(s)}{SteerIn(s)} = \frac{-11000s + 1100000}{s^4 + 161s^3 + 7215s^2 + 122500s + 1100000} \quad \text{Equation 4-3}$$

Table 4-5. Steering motor eigenvalues

Eigenvalue (rad/s)	Eigenvalue (Hz)	Eigenvalue Source
-100	-15.92	Pade Approximation of Delay
-40	-6.37	Motor Controller
-10.5+12.84i	-1.67+2.04i	Physical Motor Response
-10.5-12.84i	-1.67-2.04i	Physical Motor Response

Figure 4-9 and Figure 4-10 show the comparison of the simulated motor response to the real response. While there are visible differences at the beginning and end of the responses, the overall relationship is a close match. The overall delay in the response from the motor varies based on the commanded position as it is a combination of the non-linear region delays (about 0.2s total) and the linear region delay of 0.00138s/deg. This delay is taken into account in later analyses.

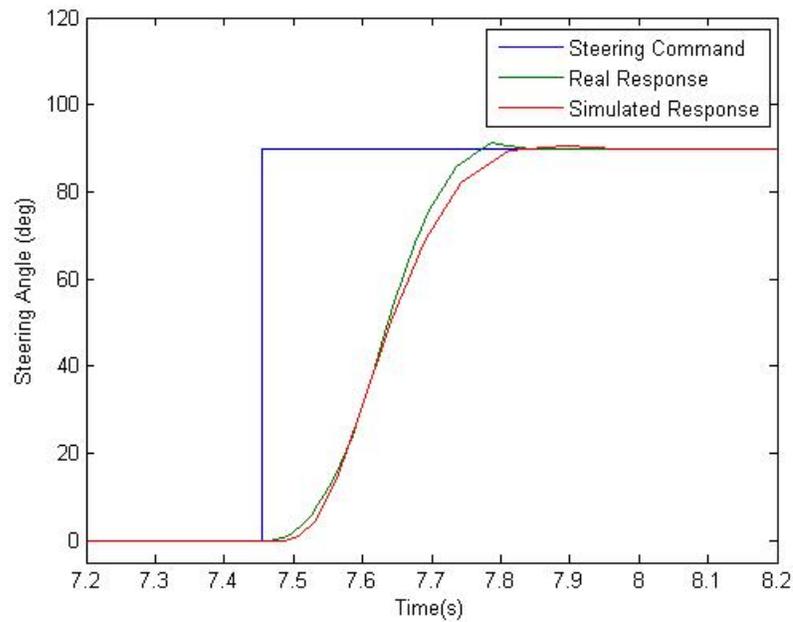


Figure 4-9. Comparison of simulated and real commanded 90° wheel angle

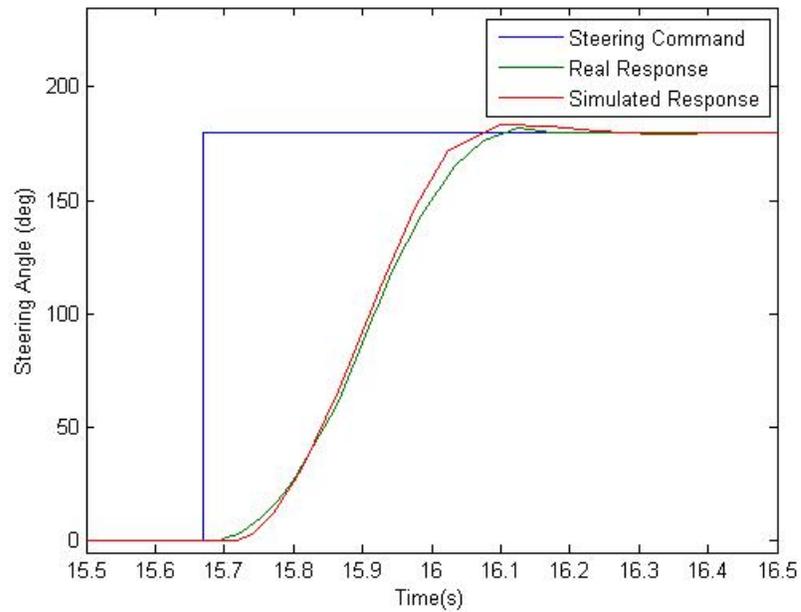


Figure 4-10. Comparison of simulated and real commanded 180° wheel angle

4.2 Test vehicle modeling

4.2.1 Bicycle model

In order to simulate the vehicle motion, a model of the vehicle must be developed. The simplest method for modeling four wheeled vehicles is the with 2 DOF Planar Bicycle Model. This model represents the vehicle as two wheels (one rear, one front) connected with a point mass in-between, as shown in Figure 4-11.

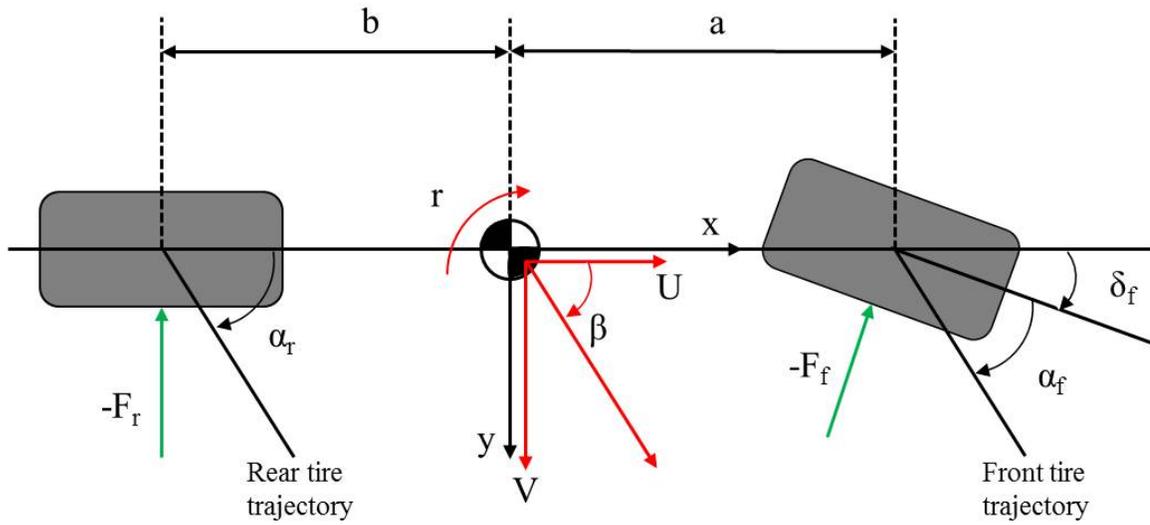


Figure 4-11. Planar Bicycle Model

A table describing the constants is shown in Table 4–6. This model was chosen because, while it neglects many dynamics of the system such as suspension deflection and body roll, it is simple to model, has only 2 degrees of freedom, and is easily linearized. The bicycle model relies on many assumptions including:

- Small angle steering
- Linear tire model
- Constant forward velocity
- No body roll
- No pitch
- No frame twist
- No suspension
- No vertical motion
- Flat road surface
- No aerodynamics

Table 4–6. List of bicycle model properties

Constant	Meaning
b	Distance from rear tire to center of mass
a	Distance from front tire to center of mass
F_r	Lateral force on rear tire
F_f	Lateral force on front tire
α_r	Rear slip angle
α_f	Front slip angle
δ_f	Front steering angle
U	Longitudinal velocity
V	Lateral velocity
r	Yaw rate
Ψ	Yaw angle
β	Sideslip angle

The bicycle model is analyzed by calculating the sum of forces in the y-direction, and a sum of moments in the z-direction.

$$\sum F_y = m(\dot{V} + Ur) = -F_r - F_f \quad \text{Equation 4-4}$$

$$\sum M_z = I_z \dot{r} = -F_f a + F_r b \quad \text{Equation 4-5}$$

The linear tire assumption and small angle approximation are implemented.

$$-F_f = C_{\alpha_f} \alpha_f \quad \text{Equation 4-6}$$

$$-F_r = C_{\alpha_r} \alpha_r \quad \text{Equation 4-7}$$

$$\alpha_f = \frac{V + ar}{U} - \delta_f \quad \text{Equation 4-8}$$

$$\alpha_r = \frac{V - br}{U} \quad \text{Equation 4-9}$$

Finally, all equations are combined and written in their state space representation with V and r as states.

$$\begin{bmatrix} \dot{V} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{C_{\alpha f} + C_{\alpha r}}{mU} & \frac{aC_{\alpha f} - bC_{\alpha r}}{mU} - U \\ \frac{aC_{\alpha f} - bC_{\alpha r}}{I_z U} & \frac{a^2 C_{\alpha f} + b^2 C_{\alpha r}}{I_z U} \end{bmatrix} \begin{bmatrix} V \\ r \end{bmatrix} + \begin{bmatrix} -\frac{C_{\alpha f}}{m} \\ -\frac{aC_{\alpha f}}{I_z} \end{bmatrix} \delta_f \quad \text{Equation 4-10}$$

4.2.2 Simulation development

Knowing the real-world specifications of the truck, the vehicle's dynamic response was modeled in MATLAB and Simulink. The vehicle specifications, state matrices, and plotting were completed in MATLAB while the remainder of the modeling was completed in Simulink. The Simulink diagram for a lane change maneuver is shown in Figure 4-12.

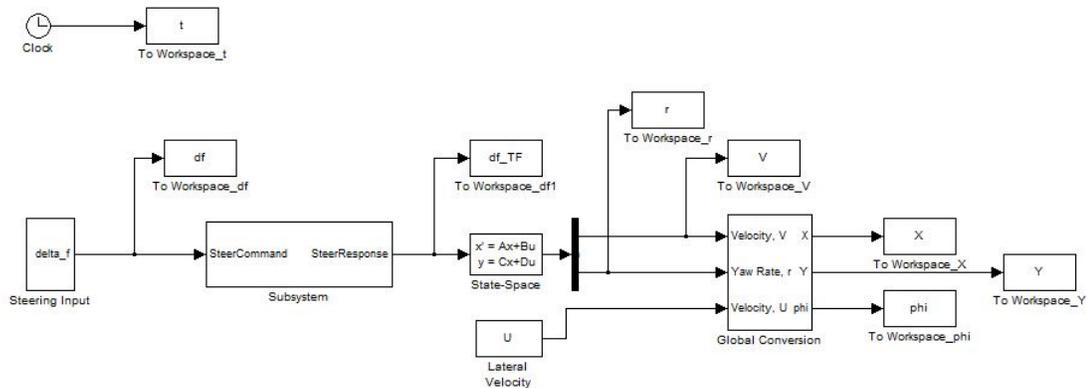


Figure 4-12. Simulink diagram of the bicycle model with steering motor response

The steering input for a lane change maneuver is a sine wave that is turned on and off after one full period using a step input. The output of the state space model is V and r , but since they are in coordinates local to the vehicle it is necessary to convert the vehicle motion to global coordinates. The relationship between the local vehicle coordinates is shown graphically in Figure 4-13 and the transformation equations are shown in Equation 4-11 and Equation 4-12.

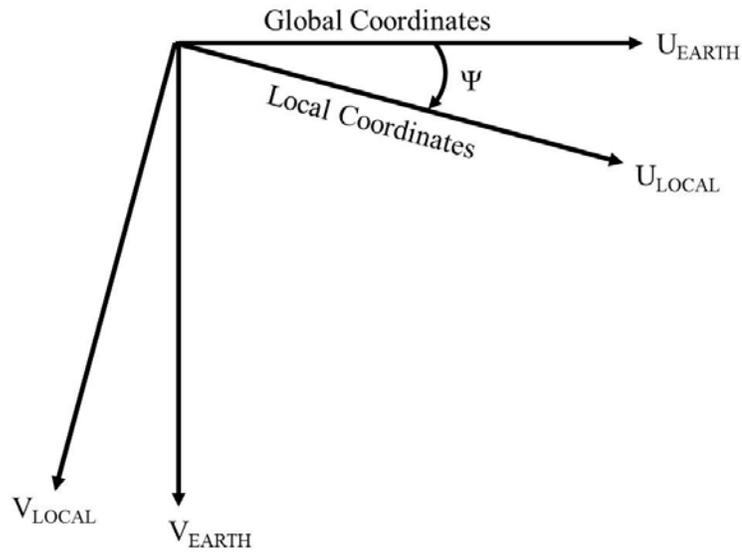


Figure 4-13. Local vehicle to global coordinate conversion

$$U_{EARTH} = U_{LOCAL} \cos \Psi - V_{LOCAL} \sin \Psi \quad \text{Equation 4-11}$$

$$V_{EARTH} = U_{LOCAL} \sin \Psi + V_{LOCAL} \cos \Psi \quad \text{Equation 4-12}$$

Using these transformations, the global response of the vehicle position was plotted for a normal lane change maneuver during manual driving without the use of the steering motor. The maneuver was simulated at 20m/s with a 0.1Hz sine wave steer command. This response is shown in Figure 4-14. The actual vehicle response of the test truck compared to the bicycle model was shown by a member of this research group to demonstrate the accuracy of the bicycle model. Figure 4-15 shows Brown's [56] comparison of yaw rate and lateral velocity for simulated and collected data during a double lane change maneuver at 16m/s.

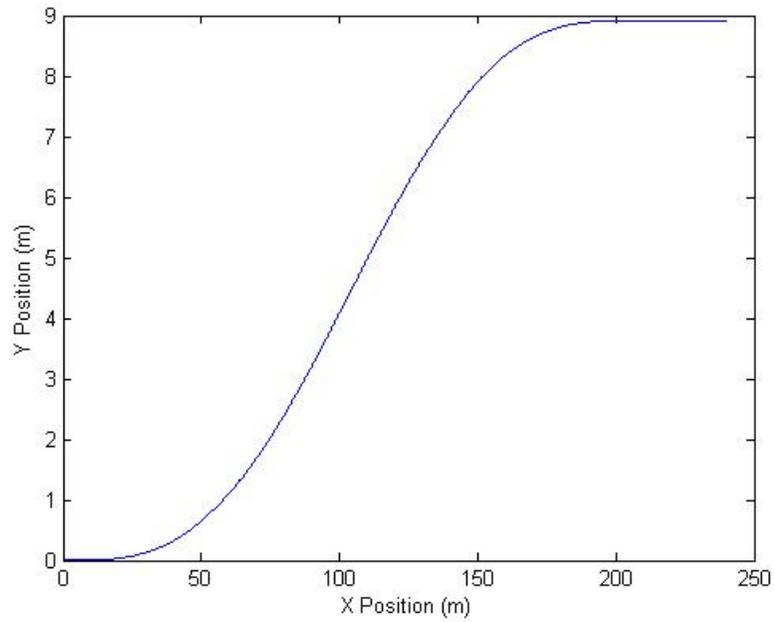


Figure 4-14. Simulated lane change maneuver for the truck at 20m/s

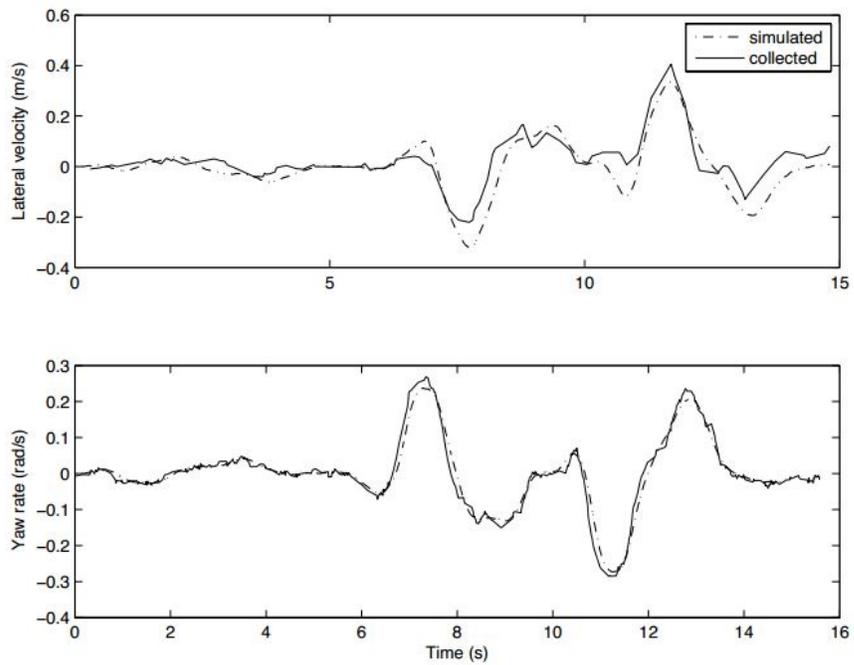


Figure 4-15. Simulated vs. real response for a double lane change at 16m/s [56]

While Figure 4-15 shows that the bicycle model provides a good representation of the real test vehicle, it does not demonstrate the effects of driving the truck with the steering motor. To properly simulate this motor response, the transfer function and rate limit for the motor developed in Section 4.1.3 were placed in the Simulink diagram between the steering command and the state space equation. These blocks were included in the Simulink diagram in Figure 4-12. Figure 4-16 shows the response of the steering motor to the steering command for a lane change maneuver

. Figure 4-17 shows the global vehicle position for the vehicle with and without the steering motor response. While the steering motor does not cause a significant delay in the system there is a finite difference of about 0.15s in vehicle response between the model with and without the motor. For this reason, the remainder of the simulations will use the steering motor model.

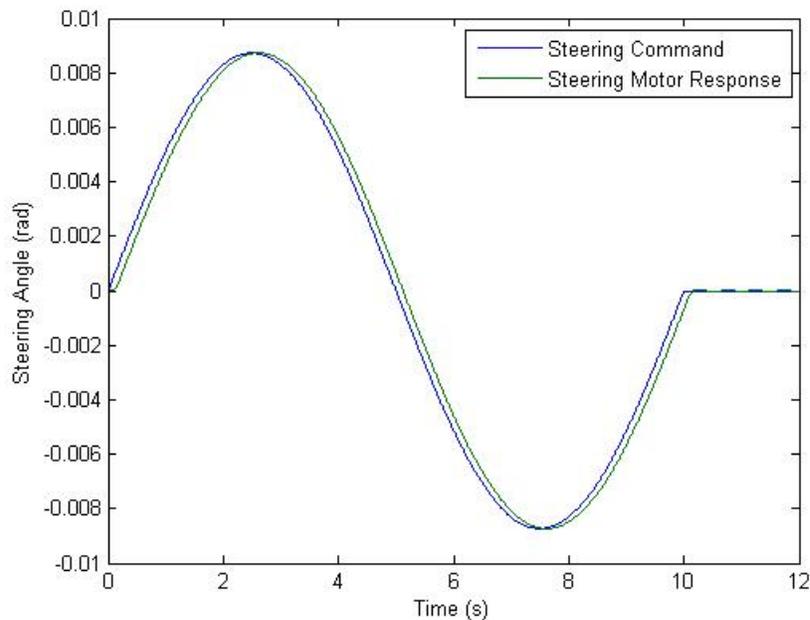


Figure 4-16. Steering command vs. steering motor response

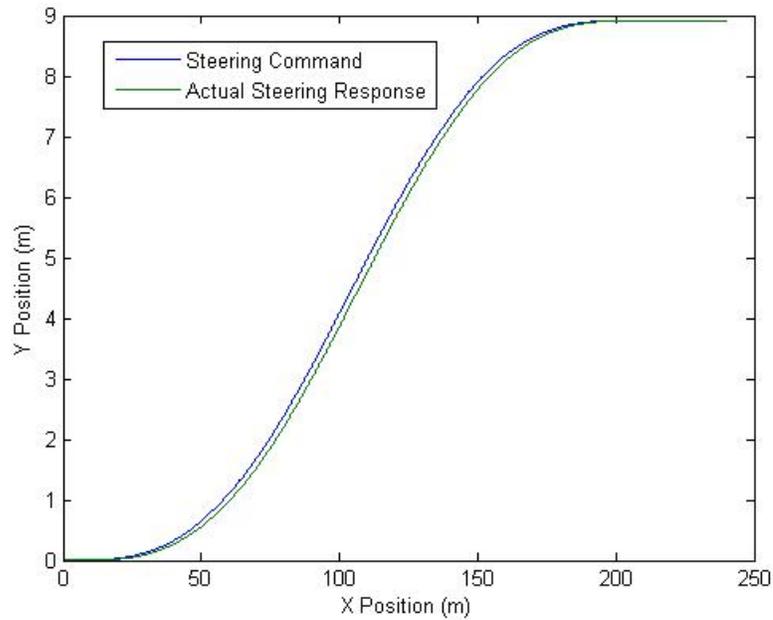


Figure 4-17. X position vs. Y position during a double lane change with motor response at 20m/s

4.2.3 Simulating a human driver

While the vehicle bicycle model is a good estimate of a vehicle's motion for a given steering input, it does not model how a human driver chooses the input based on a given path. While a number of driver models exist, this work focuses on a single point preview controller based on the optimal preview controller developed by MacAdam in 1981 [57] which is presently the most commonly used model to emulate human driver behavior in closed-loop vehicle control. The MacAdam controller uses the vehicle bicycle model and the assumption that steering angle remains constant over some preview time to find the error between a previewed input and the previewed output of the vehicle. This is shown graphically in Figure 4-18. A cost function is then applied to minimize this error.

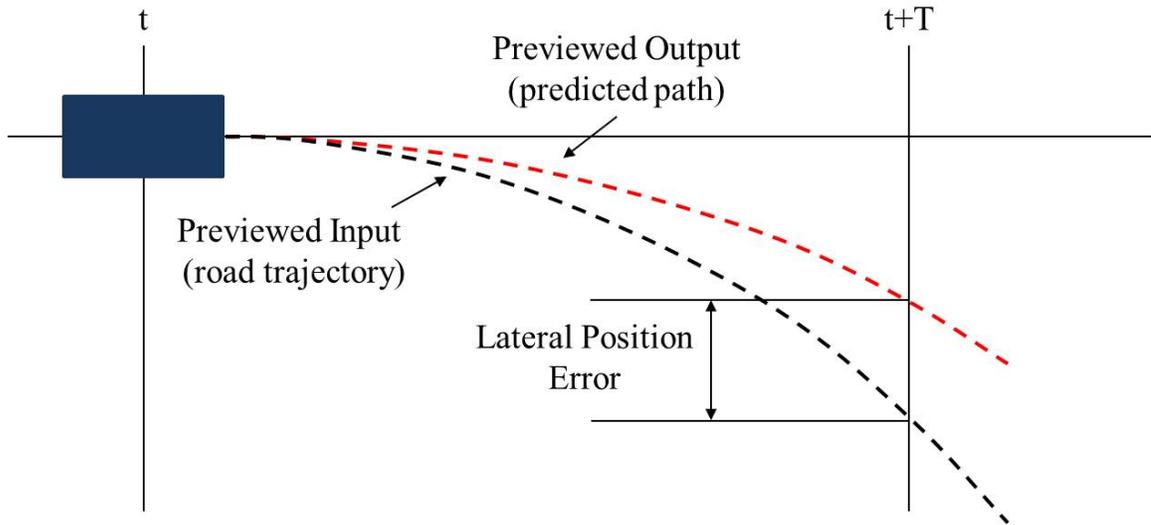


Figure 4-18. Graphical representation of the MacAdam model for lateral error

Past work by Stankiewicz [58] showed that the open loop prediction of vehicle states is calculated with Equation 4-13 where T is the preview time, $\Phi(t+T)$ is the state transition matrix approximation, u is the steering input, and Ψ is described by Equation 4-14. Stankiewicz analyzed the accuracy of various state transition matrix approximations and found that the MATLAB numerical approximation is the best, so that method was chosen for this work.

$$x(t+T) = \Phi(t+T, t)x(t) + \Psi T B u \quad \text{Equation 4-13}$$

$$\Psi = I + \sum_{k=1}^{\infty} \frac{(TA)^k}{(k+1)!} \quad \text{Equation 4-14}$$

As described by MacAdam, the closed loop controller is determined by minimizing the local performance index shown in Equation 4-15, where f is the previewed input, y is the previewed output, and W is an arbitrary weighting function [57]. The resulting optimal controller is shown in Equation 4-16 where K is defined by Equation 4-17.

$$J \triangleq \frac{1}{T} \int_t^{t+T} \{[f(\eta) - y(\eta)]W(\eta - t)\}^2 d\eta \quad \text{Equation 4-15}$$

$$u^o(t) = \frac{f(t+T) - y(t+T)}{TK} \quad \text{Equation 4-16}$$

$$K \triangleq C^T \left[I + \sum_{n=1}^{\infty} \frac{A^n T^n}{(n+1)!} \right] B \quad \text{Equation 4-17}$$

At its most basic, the MacAdam controller is simply a proportional controller on the error between a previewed road trajectory at time $t+T$ and the previewed vehicle trajectory at time $t+T$. In order for this system to be modeled in MATLAB and Simulink, the bicycle model described in Equation 4-10 must be expanded to include lateral position and yaw angle as an output. The state equation for lateral velocity is presented in Equation 4-18, resulting in the state space equation shown in Equation 4-19.

$$\dot{y} = V + U\Psi \quad \text{Equation 4-18}$$

$$\begin{bmatrix} \dot{y} \\ \dot{V} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & U \\ 0 & \frac{C_{\alpha f} + C_{\alpha r}}{mU} & \frac{aC_{\alpha f} - bC_{\alpha r}}{mU} - U & 0 \\ 0 & \frac{aC_{\alpha f} - bC_{\alpha r}}{I_z U} & \frac{a^2 C_{\alpha f} + b^2 C_{\alpha r}}{I_z U} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ V \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{C_{\alpha f}}{m} \\ -\frac{aC_{\alpha f}}{I_z} \\ 0 \end{bmatrix} \delta_f \quad \text{Equation 4-19}$$

Figure 4-19 shows the vehicle response for the closed loop MacAdam controller during a lane change at 20m/s. For this run there is no delay and 1 second of preview is used. The motor model is used in this simulation. This plot shows that the MacAdam controller effectively enables tracking of a reference road input. This model is used later to run experiments to simulate the vehicle response based on the delay or preview time provided to a driver.

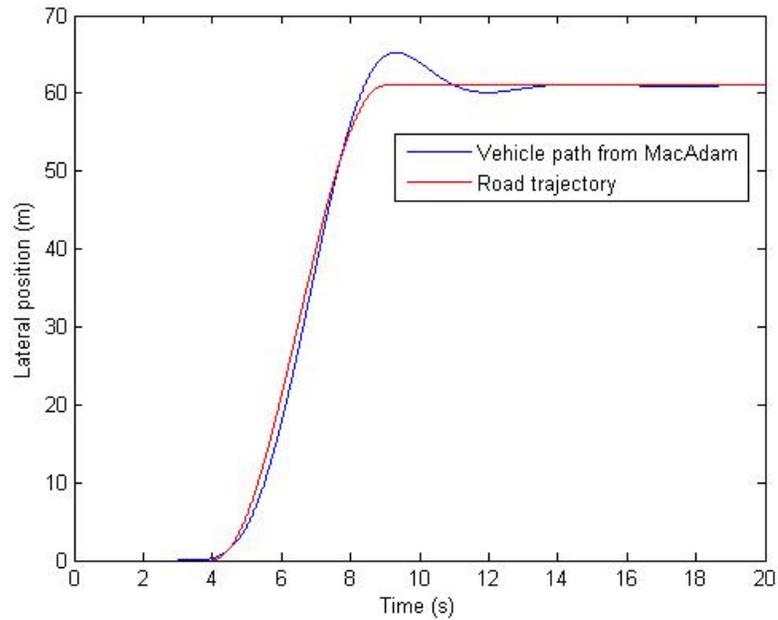


Figure 4-19. Lateral position vs. time for a lane change at 20m/s with MacAdam controller

4.3 Testing parameters

4.3.1 Test descriptions

A total of four tests were conducted in this work. Tests 1 and 2 examine the simulated response of the vehicle while varying either the preview time or system delay. Test 1 is a simulation of the vehicle response with the MacAdam controller to a lane change maneuver at a constant speed with varying preview time. The method used in this test is simple; since preview time is a direct input to the vehicle model, the phase margins are found by simply varying this input and analyzing the resulting Bode plots of the system dynamics. This test provides an expected driver phase margin for various preview times, which is then used to find the phase margin associated with various delays.

Test 2 is a simulation of the vehicle response with the MacAdam controller to a lane change maneuver at a constant speed with varying delay. The method used in this test is to add a transport delay block to the vehicle feedback and steering input in the Simulink diagram with each block representing half of the total delay in the system. The effect of the delay was slightly different if applying it to only the feedback or only the input; the full delay on the feedback slightly increased oscillations while the full delay on the input slightly decreased oscillations. The simulation was run with half of the delay on each because this is the method used in physical testing, and is believed to be the one that most accurately represents a real round-trip delay. This test identifies the delays at which the vehicle becomes oscillatory and unstable. Using this information in combination with the results from Test 1, the phase margin for the human driver with no delays is found. Subsequently, the phase margin and stability of a time-delayed teleoperated vehicle can be estimated by subtracting the phase margin associated with the delay (found in Test 1) from the human driver's phase margin (found in Test 2).

Tests 3 and 4 are the same as 1 and 2, respectively, but with the real test vehicle. Test 3 is a physical test with the human driver in the test vehicle, but teleoperating the vehicle from a visually isolated driving simulator within the cab of the vehicle with varying preview time. The visually isolated driving simulator is shown in Figure 4-20 and Figure 4-21. This test provides physical data with which Test 1 is compared, along with subjective feedback from the test subject on the effect of limited preview time on driving ability. Similarly, Test 4 is a physical test with the driver inside the vehicle in the same manner as Test 2, but with varying delay. This test provides physical data with which Test 2 is compared, along with subjective feedback from the test subject on the effect of increasing delays on driving ability. The exact methods used in each of these tests are more complicated than Tests 1 and 2 and will therefore be discussed in detail in Chapter 6.

The decision to teleoperate the vehicle from a driving simulator within the vehicle cab was deliberate. By riding in the vehicle, the test subject is still influenced by the inertial movement of the vehicle. As the MacAdam controller uses the vehicle states to determine the control gain, this method creates a more accurate representation of that controller. Additionally, inertial feedback is eliminated as a variable to provide a more accurate analysis. It is important to note, however, that haptic feedback was not provided on the steering wheel. While this does slightly affect test results, it was unavoidable as providing haptic feedback to the steering wheel through ROS has not yet been implemented.



Figure 4-20. In-vehicle driving simulator



Figure 4-21. View from driving simulator with limited preview time

4.3.2 Physical testing conditions

The test subject in this study is the human driver who controls the steering while a separate test operator onboard controls the throttle and brakes to provide both consistent speeds during testing and increased safety. For all physical tests, a human driver was present to intervene immediately since some the tests involve pushing a vehicle to the brink of stability. Additional safety protocols include an emergency stop button within reach of the test operator and test subject when he/she is in the vehicle, along with an emergency stop button on the outside of the vehicle. An additional operator is present outside of the vehicle with both a wireless engine kill switch and a wireless emergency brake that is powered independently from the rest of the vehicle. And of course, the testing area of the test track was shut down from all external vehicle traffic for these experiments.

In the physical tests, the subject was instructed to follow a 30.5m radius circular path marked on the skid pad of the Larsen Transportation Institute (LTI) test track at Penn State University. The exact GPS coordinates of this path was determined using the Novatel DGPS and LiDAR scans from past experiments. This testing area is shown in Figure 4-22.



Figure 4-22. Circular path in the testing area at LTI

CHAPTER 5

VEHICLE SIMULATION WITH VARYING PREVIEW TIME AND SYSTEM DELAYS

As discussed in the previous section, simulations were run to find the normal phase margin of a human driver and the phase margins associated with various time delays. These were used to predict the stability of the vehicle given various time delays. In this section, simulated experiments are run with MATLAB and Simulink with either no time delay and varying preview time, or constant preview time and varying time delays. By varying preview time, the phase margin of various system conditions is found and used to compute the phase margin associated with various time delays. By varying time delays, the delay at which the vehicle become oscillatory, unstable, etc. is found. The data from these two tests are then used in combination to determine a human's non-delayed phase margin and subsequently predict the stability of the vehicle based on a phase margin analysis.

5.1 Test 1 – MacAdam simulations with varying preview time

5.1.1 Test setup and results

The purpose of this simulated test is to determine the phase margin of the vehicle at various preview times. For each of these preview times, the phase margin of the system was calculated using the MATLAB function “margin” on the closed loop system to provide a numerical analysis of stability. In order to find the phase margin including the steering motor dynamics, the state space bicycle model is augmented with the transfer function for the steering motor. The A , B , C , and D matrices for the full augmented state space equation for the vehicle

including motor dynamics are shown in Equation 5-1 through Equation 5-4. The derivation of these state space matrices is shown in “Appendix A – Derivation of the Bicycle Model with the Motor Model”.

$$A = \begin{bmatrix} 0 & 1 & 0 & U & 0 & 0 & 0 & 0 \\ 0 & \frac{C_{\alpha f} + C_{\alpha r}}{mU} & \frac{aC_{\alpha f} - bC_{\alpha r}}{mU} - U & 0 & 0 & 0 & 11000\left(\frac{C_{\alpha f}}{m}\right) & -1100000\left(\frac{C_{\alpha f}}{m}\right) \\ 0 & \frac{aC_{\alpha f} - bC_{\alpha r}}{I_z U} & \frac{a^2C_{\alpha f} + b^2C_{\alpha r}}{I_z U} & 0 & 0 & 0 & 11000\left(\frac{aC_{\alpha f}}{I_z}\right) & -1100000\left(\frac{aC_{\alpha f}}{I_z}\right) \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -161 & -7215 & -122500 & -1100000 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{Equation 5-1}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 5-2}$$

$$C = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -11000 \ 1100000] \quad \text{Equation 5-3}$$

$$D = 0 \quad \text{Equation 5-4}$$

In order to run a closed loop analysis, the feedback equations for the MacAdam controller must be derived. The closed loop equations for the MacAdam controller with the motor model and without the motor model are show in Equation 5-5 and Equation 5-6, respectively. The full derivation of these equations is performed in “Appendix B – Derivation of the Feedback Equations for the MacAdam Controller”

$$\dot{x} = (A_m - B_m K^* C_m - B_m K^* D_m C_s)x + (B_m K^*)y_{in} \quad \text{Equation 5-5}$$

$$y = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]x$$

$$\dot{x} = \left(A_o - \frac{B_o K^* C_o}{1 + K^* D_o} \right) x + \left(\frac{B_o K^*}{1 + K^* D_o} \right) y_{in}$$

Equation 5-6

$$y = [1 \ 0 \ 0 \ 0]x$$

To verify the accuracy of the state-space representations of the vehicle and controller, the vehicle model with the nonlinear block-diagram representation of the steering and vehicle dynamic were compared to the above linear state-space representation of the same dynamics. The comparison of these two simulations in Figure 5-1 shows a very close fit between the two model representations.

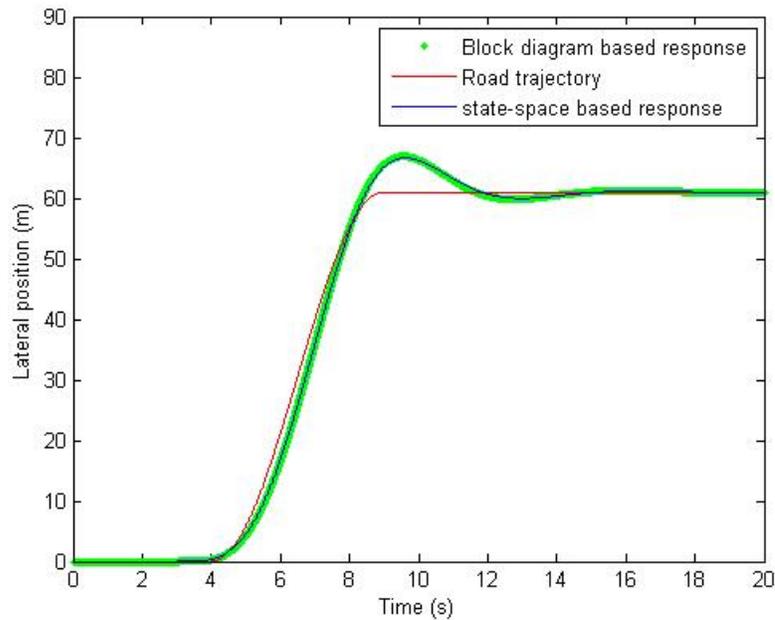


Figure 5-1. Comparison of block diagram based motor model to state space based motor model

Since systems become unstable at 0° phase margin (and for practical purposes are quite difficult to control below 30° of phase) these data provide the precise point where the vehicle should become unstable. In a swept-sine analysis, there is a fixed relationship between the phase angle, frequency, and system delay given in Equation 5-7.

$$t_x = \frac{\varphi}{\omega}$$

Equation 5-7

From the phase margin and crossover frequency, the amount of delay associated with each preview time is calculated. This represents, according to a frequency response, the amount of delay that the system should be able to tolerate prior to becoming unstable. This is an idealization that assumes a sine wave input. However, the MacAdam controller assumes a constant input over the preview horizon, and thus the data must be interpreted with suspicion – especially for long preview horizons where even the Pade approximation may be invalid.

An example Bode plot of the closed-loop system with 0.3 seconds of driver preview is shown in Figure 5-2 for a system without actuator dynamics and Figure 5-3 with actuator dynamics. The phase margins are shown in Table 5–1 for the simulation without the motor model and in Table 5–2 for the simulation with the motor model. The highlighted boxes show unstable responses. For all preview times in Table 5–1, the allowable delay is larger than the preview time, e.g. the controller is able to give more phase benefit by using preview than phase is taken away if the same delay were added. However, with the actuator dynamics in Table 5–2, this benefit is only occurring for preview times greater than 0.5 seconds.

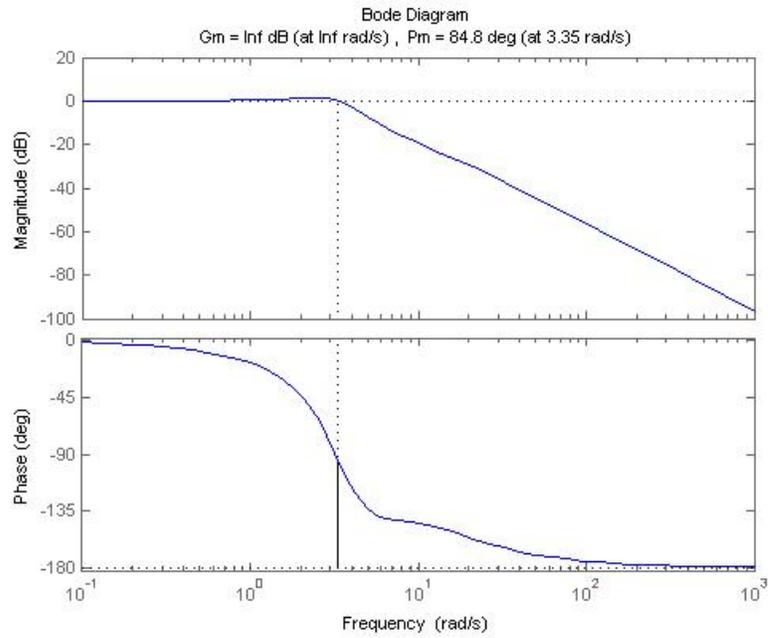


Figure 5-2. Bode plot of the closed-loop system with 0.3 seconds of driver preview without actuator

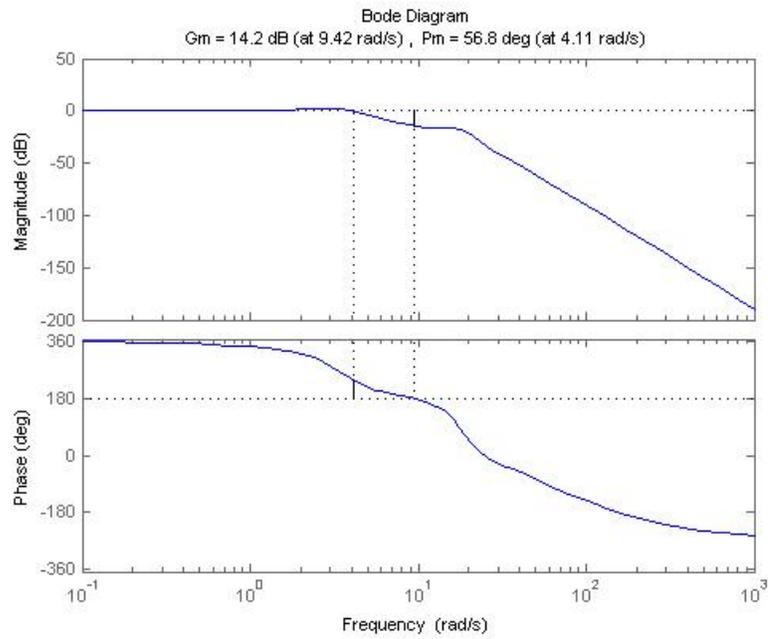


Figure 5-3. Bode plot of the closed-loop system with 0.3 seconds of driver preview with actuator

This information is useful for finding a human's normal phase margin with the data from the varying system delay analysis in Section 5.2. For example, if an experiment shows that a human-driven vehicle becomes unstable with 0.3 seconds of delay added to the human's response (as later experiments show) then Table 5–2 shows that the human controller must be “designed” to give roughly 61 degrees of phase margin.

Table 5–1. Phase margin for simulated preview time test without steering motor model

Preview Time (s)	Phase Margin (deg)	Frequency (rad/s)	Calculated Delay (s)
4.00	89.4	0.25	6.19
3.00	89.2	0.34	4.62
2.00	88.8	0.50	3.04
1.00	87.7	1.03	1.49
0.50	85.7	2.06	0.73
0.40	85.1	2.56	0.58
0.35	84.8	2.87	0.52
0.30	84.8	3.35	0.44
0.25	85.5	3.95	0.38
0.22	86.5	4.40	0.34
0.20	87.7	4.77	0.32
0.17	90.3	5.44	0.29
0.15	92.9	6.01	0.27
0.12	98.4	7.16	0.24
0.10	82.8	7.84	0.18
0.07	43.6	6.67	0.11
0.05	57.7	9.14	0.11

Table 5–2. Phase margin for simulated preview time test with steering motor model

Preview Time (s)	Phase Margin (deg)	Frequency (rad/s)	Calculated Delay (s)
4.00	87.6	0.26	5.97
3.00	86.8	0.35	4.39
2.00	85.2	0.53	2.83
1.00	80.0	1.10	1.27
0.50	69.1	2.34	0.56
0.40	63.9	3.00	0.37
0.35	61.0	3.42	0.31
0.30	56.8	4.11	0.24
0.25	52.5	5.03	0.18
0.22	49.6	5.82	0.15
0.20	47.1	6.54	0.13
0.17	39.0	8.39	0.08
0.15	-104.0	16.7	-0.11
0.12	-155.0	17.7	-0.15
0.10	-102.0	13.8	-0.13
0.07	-56.8	7.40	-0.13
0.05	-89.4	9.99	-0.16

To confirm that the phase margins give results that make physical sense, simulations were performed in MATLAB and Simulink by running the MacAdam controller and vehicle models developed in Section 4.2.3 with various preview times from 0.05s to 4s. The speed of the vehicle is independent of preview time so vehicle speed was held to a constant 10m/s, (22 mph) which is considered a safe speed with which to run the physical tests at the test track. As system stability is independent of vehicle trajectory, a simple lane change maneuver was used for this analysis. The response to this lane change maneuver at some of the tested preview times is shown in Figure 5-4 to show how tracking error varies. The data shows that instabilities emerge with around 0.15s or less preview, which is in complete agreement with the phase analysis presented in Table 5–2.

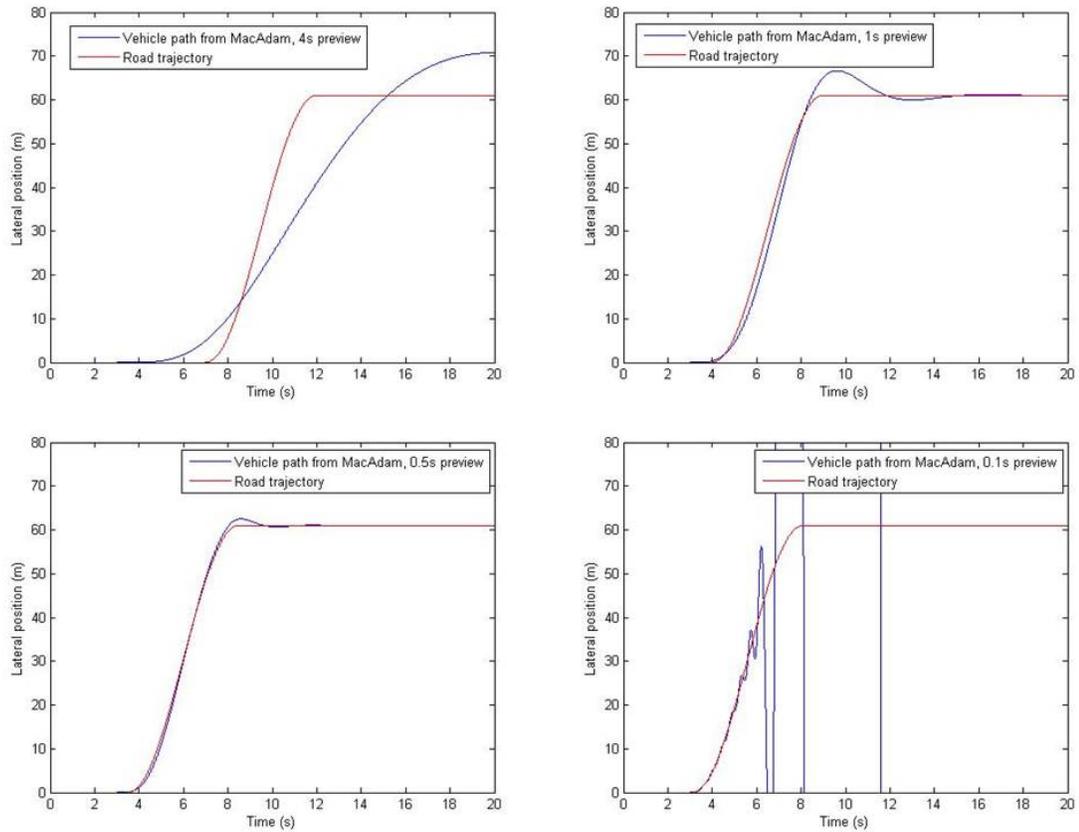


Figure 5-4. Example results for a lane-change maneuver for MacAdam controller for various preview times

5.1.2 Discussion

The data shows that a vehicle enters an unstable condition (less than 0° phase margin) with around 0.15s or less preview time. In terms of the effect of limited preview time on a human driver's performance, this data shows that humans are theoretically able to operate a vehicle with very limited preview time. For example, at 10m/s this is only 1.5m in front of the driver, which is not even within the visible range of the driver: the hood of the vehicle usually prevents the view of anything within about 5m on the ground. This implies that drivers are mostly using the visible

road geometry to infer the vehicle's position just in front of the bumper (but not visible), and then controlling the vehicle around this position.

To drive fast enough that the look-ahead point is actually visible to the driver, a typical vehicle must be driven at over 50m/s to achieve an unstable condition. This is both extremely unsafe and impractical in this test vehicle (and most vehicles in general). Interestingly, it is known that most drivers require special training for high-speed driving, which is generally understood to occur at speeds above 50 m/s (e.g. race-car driving schools).

It is unsurprising that phase decreases much more rapidly for the system with the steering motor model than in the system without it due to the delays inherent in the motor. It is interesting, however, that in the system without the motor model phase margin does not decrease much, and increases in some cases. By inspecting the Bode plots, it is clear that the model without the steering motor is limited within 180° phase margin while the model with the steering motor has much greater boundaries, allowing much more significant changes in phase margin.

A separate, but interesting, result of this simulation is that the driver will track the road trajectory with less error in cases with less preview distance. This is a counterintuitive result: that drivers perform better at tracking with more limited view. This effect is due to several factors: for example the driver will “cut the turn” more with larger and larger preview. Further, the simplistic driver model used here, while analytically tractable, is not completely representative of how human drivers extrapolate lane information. It is likely that drivers use an array of preview points instead of a single preview point as simulated by the single point MacAdam model. Additionally, human drivers likely use preview for different purposes. For example, lane tracking appears to occur at shorter distances, but collision avoidance maneuvers are likely planned at longer distances. Long distance preview may also be more important during high performance driving such as racing.

5.2 Test 2 – MacAdam simulations with varying time delays

5.2.1 Test setup and results

The purpose of this simulated test is to determine the stability of the vehicle at various delays, for a constant preview time. Simulations were run in MATLAB and Simulink by running the MacAdam controller and vehicle models developed in Section 4.2.3 with various delays from 0.1s to 0.4s. As discussed in Section 4.3.1, this was done simply adding a transport delay block to both the feedback and steering input, each representing half of the total delay. The vehicle speed was held to a constant 10m/s, (22 mph) which is considered a safe speed with which to run the physical tests at the test track. As noted by Ungoren, 0.5s to 2s of preview is used by most human drivers [59]. In order to represent the human driver accurately and provide low tracking error, a 0.5s preview time was used for the simulation. As with Test 1, a simple lane change maneuver was used for this analysis.

The expected result of this test is a specific delay time that corresponds to the onset of vehicle instability. This delay time is then used in conjunction with the data presented in Section 5.1 to determine the normal phase margin of a human driver which in turn provides the ability to predict instability in vehicles based solely on phase margin. The response to this lane change maneuver at some of the tested delays is shown in Figure 5-5 to show how tracking error varies. These plots show that instabilities emerge just over 0.3s delay.

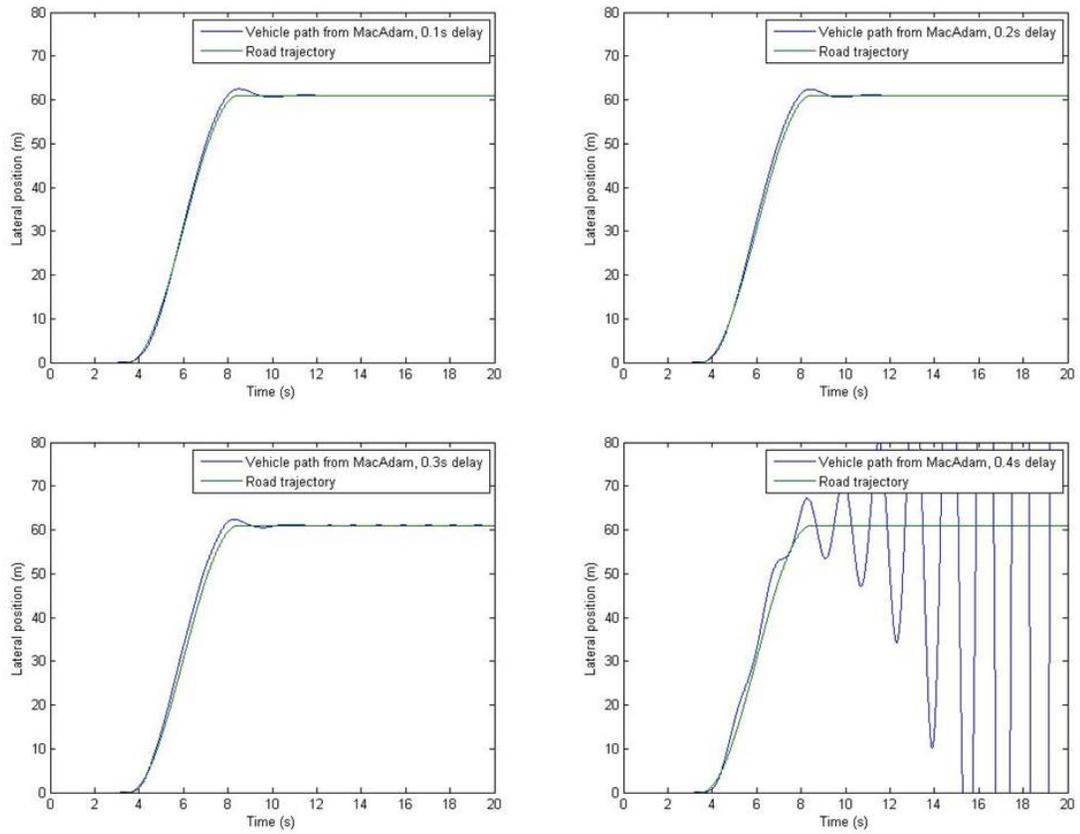


Figure 5-5. Example results for a lane-change maneuver for MacAdam controller for various delays

Through simple visual inspection of the vehicle response plots, the stability condition (stable, oscillatory, or unstable) for various time delays was found. Table 5-3 provides a summary of these stability conditions.

Table 5–3. Vehicle stability condition for various time delays

Delay (s)	Stability Condition
0.1	Stable
0.2	Stable
0.23	Stable
0.24	Stable
0.25	Oscillatory
0.26	Oscillatory
0.27	Oscillatory
0.28	Oscillatory
0.29	Oscillatory
0.3	Oscillatory
0.31	Unstable
0.32	Unstable
0.33	Unstable
0.34	Unstable
0.35	Unstable
0.4	Unstable

Knowing that the vehicle enters an unstable condition at 0.31 seconds, the phase margin of a human driver can be found in this by working backwards through Table 5–2, looking for the calculated delay closest to 0.31s, and then finding the corresponding phase margin which is 61°. Since instability occurs at about a 0° phase margin and the destabilizing delay removes about 60° of phase, this suggests that human drivers naturally operate around a 60° phase margin if they are using 0.5 seconds of preview in a MacAdam type controller.

Similarly, the point of oscillation begins at 0.25s, which corresponds to a 57° phase margin. This makes sense, as removing 57° of phase from the normal 60° of the human driver will place the vehicle just before the instability point at about 3°.

5.2.2 Discussion

The maximum phase margin of the MacAdam controller was found to be around 90° as preview is increased substantially, which is an unsurprising outcome as this is a common phase

margin of stable human systems – e.g. a critically damped response. However, this required significant preview, so much so that the trajectories were cutting corners during curves and lane changes. As a human driver would not do this, it is expected that the driver would be controlling the vehicle using a previewed position closer to the vehicle than implied by a 90° phase margin.

It was also seen that the system model without steering dynamics gives much higher phase margins than system models with steering dynamics. However, the model without an actuator dynamic (but with human in the loop) should actually include neuromuscular dynamics. Models for these dynamics are very likely to be similar to the dynamics of the electric steering actuator.

More importantly, in terms of vehicle stability this data provides a clear indication of the phase margin associated with various delays. An initial estimate based on the 0.31s instability point and Table 5–2 would suggest that human drivers typically operate at around 60° phase margin. Knowing that a system goes unstable at around 0° phase margin, this means that any delay with over 60° phase margin will cause instabilities in the system.

With steering actuator dynamics included, oscillation in the tracking response begins to occur at around only 0.25s of delay when traveling at 10m/s. The results of Test 1 suggest that the preview time needed to cause instability is difficult to experimentally validate, since the driver's preview point may be below the visible area of the driver. However, given the latencies observed in some of the hardware already, it appears plausible for a remotely-operated by-wire system to reach 0.25s of delay at these speeds. For higher speed operations, the destabilizing delay is significantly less. As shown in Figure 5-6, when traveling at highway speeds (25m/s) the point of destabilization occurs at only 0.15s and reaching instability at these speeds can be highly dangerous, if not deadly. These test results indicate that, in teleoperated vehicles, transmission delays of data is a primary concern for both system performance and, even more importantly, safety.

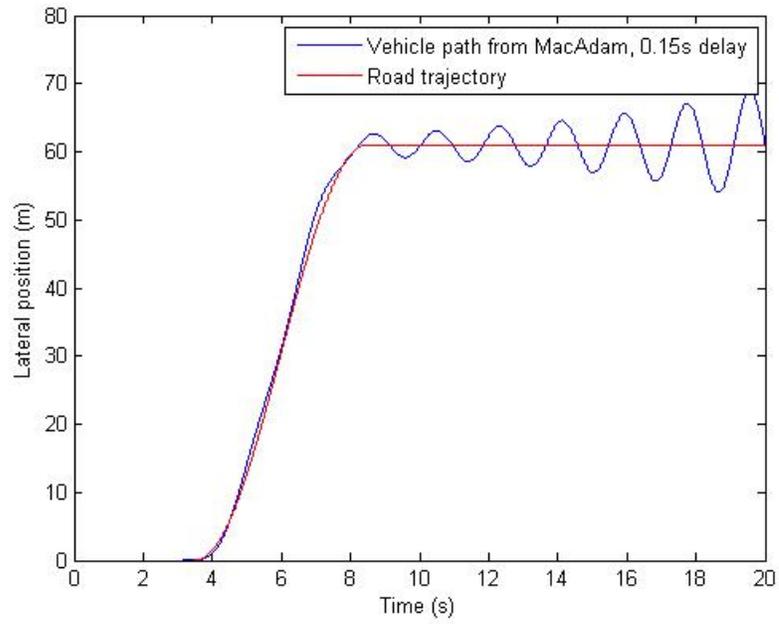


Figure 5-6. Lane change maneuver at 25m/s with 0.15s delay

CHAPTER 6

PHYSICAL TESTING WITH VARYING PREVIEW TIME AND SYSTEM DELAYS

This chapter presents an analysis of Tests 3 and 4 which examine the effect of varying preview time with a constant delay, and varying delay with a constant preview time, respectively. Using the test vehicle and setup described in Chapter 4, the human driver teleoperated the vehicle while either camera data provided to the driver was artificially limited, or delays were artificially added to the system. The methodology for each test is discussed more specifically below. These tests provide a real world comparison and application of the results found in Chapter 5 by which further understanding of the effects of delays on the system are analyzed.

6.1 Test 3 – Physical tests with varying preview time

6.1.1 Test setup

The goal of this experiment is to examine the physical results from varying preview time with a constant delay on the vehicle's stability. In each test, the driver's preview *distance* is artificially limited and, by driving at a constant speed, this distance is converted to a preview *time*. As described in Section 4.3, the human driver was seated inside the cab within the blacked out passenger seat and controlled the steering with the Logitech racing wheel based on the feed from the camera. In order to artificially modify the driver's preview distance, the image from the camera feed provided to the driver was altered via Python script to block the top of the screen, as

shown in Figure 6-1. This distance was manually calibrated by physically measuring the distance in front of the driver on the ground and generating a fit equation that translates number of pixels to view distance. The camera was securely fixed in place for all tests to ensure uniform preview distance measurements.



Figure 6-1. Comparison of full camera view to limited camera view

For each run, the speed was incrementally increased between 2.5m/s to 10m/s to obtain multiple preview time data points. As the speed was increased the test subject was in direct communication with the vehicle operator and was able to stop the test if he/she felt that the vehicle was reaching instability. Any requested stop was considered to be the point at which the vehicle reached instability.

To understand normal driver performance, the human driver was first required to follow the path in a counter-clockwise direction via teleoperation without restricted preview distance, and then with 30m, 25m, 20m, 15m, and 10m of preview distance. During each of these runs the vehicle's global position was recorded with the Novatel GPS while steering commands and feedback were recorded through the steering motor controller and the string potentiometer feedback at the steering rack.

The expected result from this experiment is that, as preview time is decreased, the driver's path tracking improves, to a point very close to the vehicle, after which further reduction of preview will reach a point of instability. The driver's path tracking is examined by calculating the driver's RMS error from the path. However, based on the results of the simulated study in Test 1, it is unlikely that the vehicle reaches instability, or even oscillations at the safe testing speeds.

6.1.2 Results

First, the overall tracking of the path for each of these tests was recorded to determine if an unstable or oscillatory condition was ever reached. Ultimately, none of the tests resulted in an oscillatory or unstable condition. An example of this overall path tracking is shown in Figure 6-2. All preview distances tested resulted in very similar tracking paths. The only variations that occurred were brief deviations from the path similar to the one seen in Figure 6-2. This variation occurred during acceleration from one speed interval to another and therefore does not appear in the RMS error calculations. While it does not appear in the RMS calculations, it is interesting in that, while this study examines driver performance at constant speeds, accelerations may cause destabilization more quickly than simply driving at constant speeds.

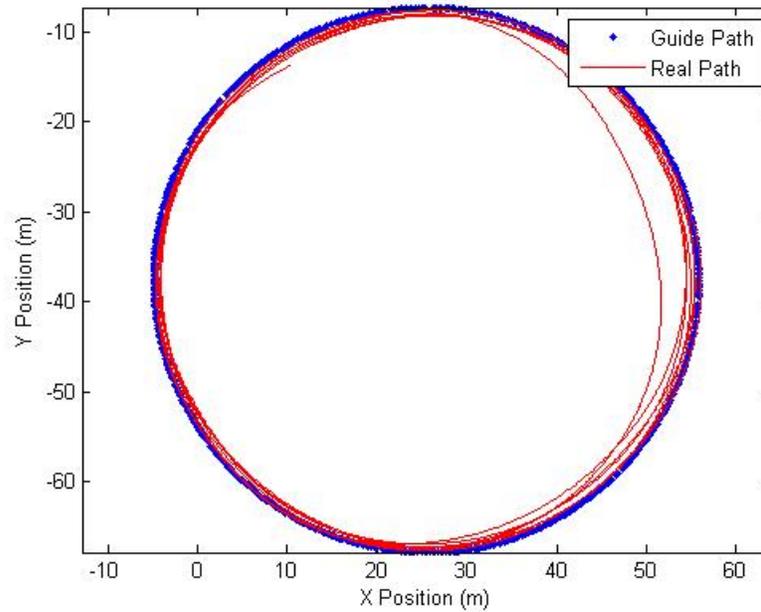


Figure 6-2. Real vehicle path vs. guide path for a 25m preview distance

For each preview distance, the individual speed intervals were analyzed to determine preview time and RMS error. Preview time was determined by dividing the preview distance by the average velocity across a nearly constant speed.

RMS error was calculated by using the built-in MATLAB function “knnsearch” that finds the nearest neighbors between two sets of data, and then returns the index of those points and the error distance between them. Using this error distance, the RMS is calculated with Equation 6-1. The RMS error is plotted in Figure 6-3 and the full results from the varying preview time tests are shown in Table 6-1.

$$RMSerror = \sqrt{\text{mean}(error^2)} \quad \text{Equation 6-1}$$

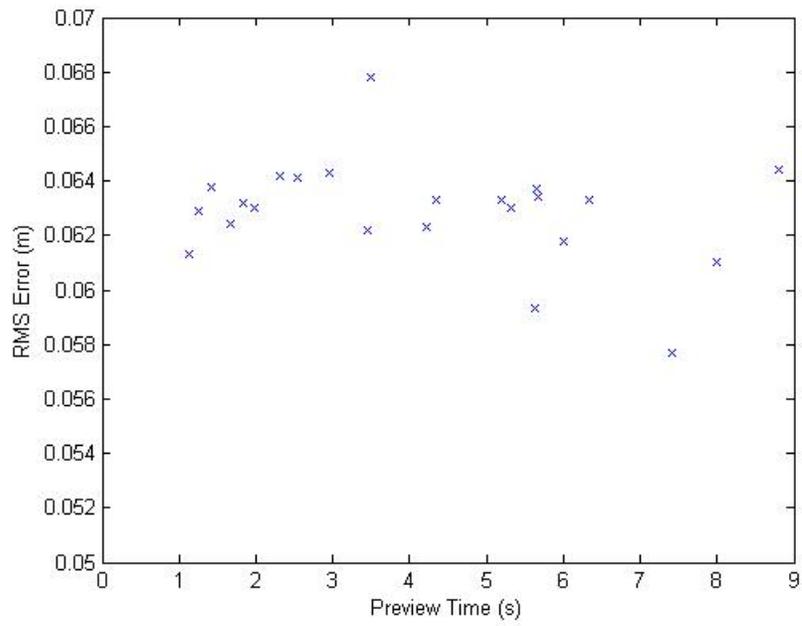


Figure 6-3. Root mean square error vs. preview time

Table 6–1. Results from varying preview time experiment

Test	Preview Time (s)	RMS Error (m)
Full View	8.81	0.0644
	7.99	0.061
	6.34	0.0633
	5.67	0.0634
30m	7.41	0.0577
	5.66	0.0637
	4.22	0.0623
	3.46	0.0622
25m	6	0.0618
	5.31	0.063
	4.35	0.0633
	3.49	0.0678
20m	5.62	0.0593
	2.95	0.0643
	2.54	0.0641
	5.19	0.0633
15m	2.32	0.0642
	1.99	0.063
	1.66	0.0624
10m	1.83	0.0632
	1.43	0.0638
	1.25	0.0629
	1.14	0.0613

6.1.3 Discussion

The results from this experiment agree with the simulated results. The smooth path tracking plots such as Figure 6-2 suggest that the vehicle never entered, or even when near an unstable condition, even when driving with only 1s preview time. The driver did noticeably leave the path at points, but recovered quickly. The uniformity of the RMS error suggests that these deviations from the path are simply due to driver error. Additionally, these deviations from the

path primarily occur during accelerations from one speed interval to another, indicating that stability is easier to maintain during constant velocity maneuvers.

RMS error trends upwards slightly as preview time decreases, but only by a few centimeters. Overall the RMS error remains mostly constant throughout the experiment. This result shows some agreement with the simulated results. The results match in that the vehicle does not go unstable for the preview time tests, but they do not match in that simulation would suggest that RMS error actually trends downwards as preview time decreases. The discrepancy between the simulated and actual results support the conjecture that human drivers, even when provided full view, control the small perturbations in vehicle path tracking off of very short preview distances while using long distance features for advanced path planning and collision avoidance.

It is also notable that the RMS error remains around 6cm, which is about the width of a lane line. This result is important because it provides a standard by which automated vehicles should be expected to adhere during self-guidance. Essentially, if a human driver is satisfied operating the vehicle within about 6cm of lateral error, then a robot should also be expected to follow the same standards.

From a subjective standpoint, the test subject found that, while driving with limited preview times was feasible, it was much more physically straining to do so than to drive with full view. A few possible conclusions can be drawn from this. First, it is possible that the additional strain is caused by innate fear of possible impending collisions that are blocked from view. Even though the test operator is able to prevent collisions and the testing environment is safe, a subconscious fear of the unknown may still exist. Second, this result supports a theory that humans actively scan the full range of view at all times for input and, when view is limited, attempt to gather more information from a limited environment which causes increased strain on the driver.

In the future it would be informative to perform this experiment in a similar manner, but by blocking out the top and the bottom of the screen and providing only a small sliver of preview to the test subject. It would be expected that the path tracking results would more closely follow those of the single point preview MacAdam controller.

Overall, this experiment indicates that drivers are able to adapt to, or even normally operate a vehicle with exceptionally limited preview time. This indicates that human drivers will not likely enter an unstable condition for typical road behaviors based on limited preview time.

6.2 Test 4 – Physical tests with varying delay

6.2.1 Test setup

The goal of this experiment is to examine the physical results from varying delays with a constant preview time on the vehicle's stability. To accomplish this, the same test setup as in Test 3 is used with the human driver in the blacked out passenger seat of the test vehicle. In this test, however, there are no changes made to the image presented to the driver. Instead, a delay is added to the image coming into the test subject while the same delay is added to the steering commands output from the test subject. Together, these two delays form the total round-trip delay for the system.

In order to create the delay in the system, a buffer was implemented on both the incoming and outgoing messages equal to half the delay. As either the camera image or steering commands were received by ROS, the data was stored in the buffer along with the timestamp of when it was received. In each iteration cycle of the code, the current time is compared to the timestamp of the oldest message in the buffer. If this oldest message is older than the delay then it is output to the

test subject (camera image) or the steering motor (steering command). A comparison of the steering command to the delayed steering command is shown in Figure 6-4.

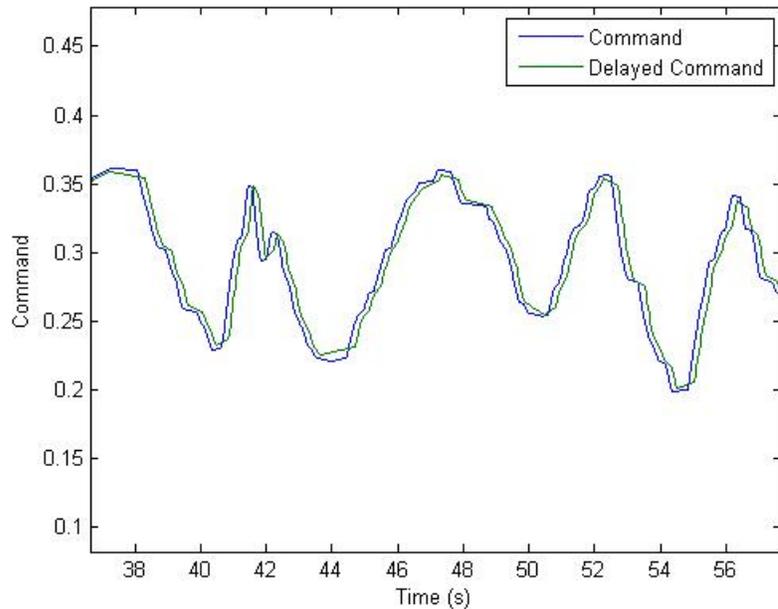


Figure 6-4. Actual command vs. delayed command in a 0.2s one-way delay

As with Test 3 the speed was incrementally increased between 3.5m/s to 10m/s to obtain multiple preview time data points during each run. Again, as the speed was increased the test subject was in direct communication with the vehicle operator and was able to stop the test if he/she felt that the vehicle was reaching instability. Any requested stop was considered to be the point at which the vehicle reached instability.

The human driver was also required to follow the same circular path in a counter-clockwise direction via teleoperation without a delay, and then with a 0.05s, 0.1s, 0.15s, and 0.02s one-way delay. During each of these runs the vehicle's global position was recorded with the Novatel GPS while steering commands and feedback were recorded through the steering motor controller and the string potentiometer feedback at the steering rack.

The expected result from this experiment is that, as delay is increased, the driver's path tracking decreases until instability is reached. The driver's path tracking is examined by calculating the driver's RMS error from the path. From the results in Section 5.2 it is expected that instability will occur around 0.3s of round-trip delay, or 0.15s of one-way delay, for a speed of 10m/s.

It is important to note that a 0.15s delay must be added to all of the artificial round-trip delays added to the system to account for the delay in the steering motor.

6.2.2 Results

As with Test 3, the overall tracking of the path for each of these tests was recorded to determine if an unstable or oscillatory condition was ever reached. As predicted, the system did reach instability, but did so at delays slightly higher than predicted by the simulation. Instability was reached with both a 0.45s (0.3s without steering motor) round-trip delay and a 0.55s (0.4s without steering motor) round-trip delay, and occurred at speeds below 10m/s. The overall path tracking for these tests are shown in Figure 6-5 and Figure 6-6. In some cases, the vehicle is able to return to the path after wild excursions; however, this is considered unstable because the vehicle would leave the road in real driving scenarios. All other tests showed results more similar to Figure 6-2 with very small RMS error.

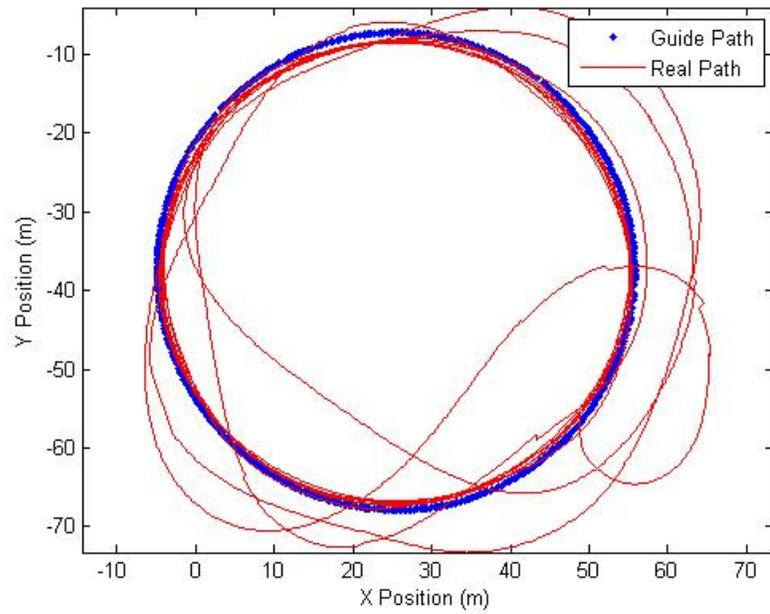


Figure 6-5. Real vehicle path vs. guide path for a 0.3s round-trip delay

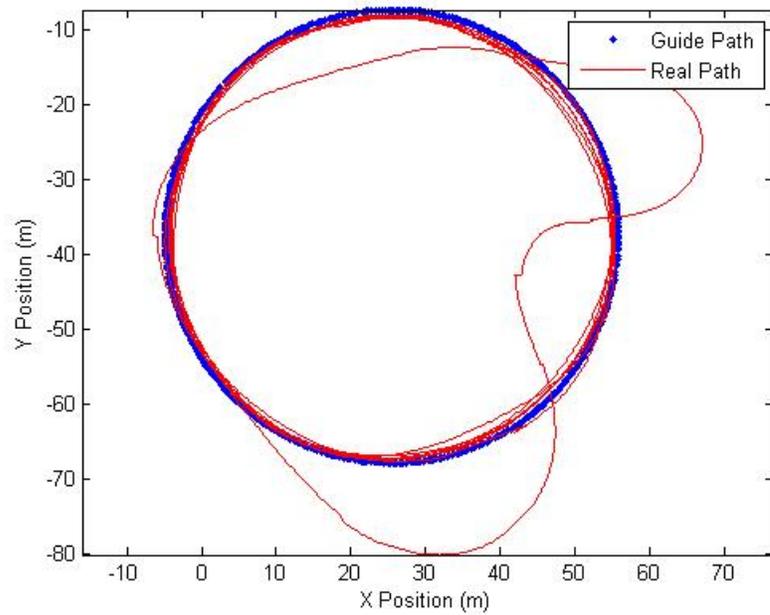


Figure 6-6. Real vehicle path vs. guide path for a 0.4s round-trip delay

In these figures, it appears that a 0.45s delay is actually more unstable than a 0.55s delay. This statement, however, is not completely proven since in the 0.55s test the speed of the vehicle was immediately dropped upon reaching instability, as requested by the test subject (and safety driver). This is shown in Figure 6-7 where the red circle identifies the point where the vehicle went unstable. On the other hand, in the 0.45s test, the test subject requested that the speed be maintained in order to regain stability, but ultimately asked to drop the speed slightly in order to get back on the track. This is shown in Figure 6-8 where the red circle again marks the point of instability. The ability of the driver to continuously cross the track at higher speeds and regain stability when returned to a lowed speed indicates that this could be considered oscillatory over fully unstable.

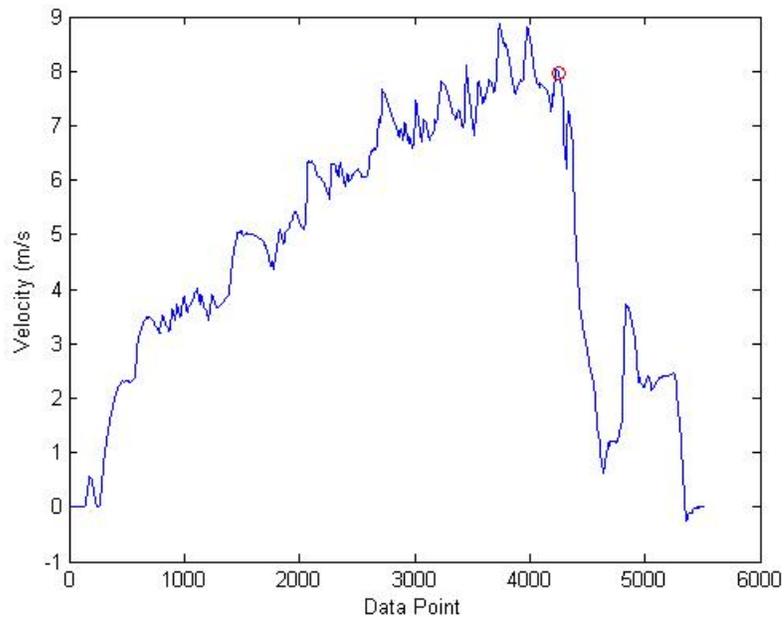


Figure 6-7. Vehicle velocity during 0.55s delay

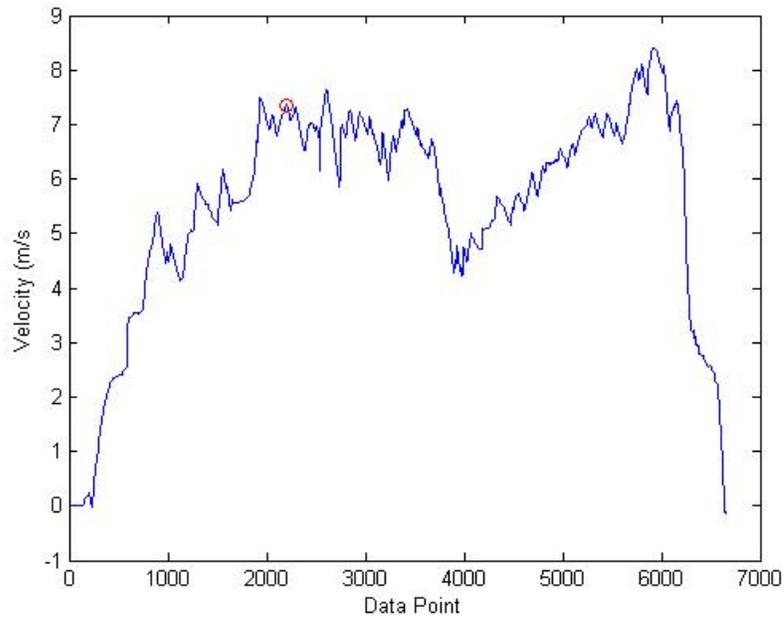


Figure 6-8. Vehicle velocity during 0.45s delay

For each delay, the individual speed intervals were analyzed to determine preview time and RMS error. RMS error was calculated in the same manner as in Test 3, only these data are grouped by delay to show how speed increased instability at each delay time. The full results are shown in Figure 6-9, with a zoomed plot of the stable points shown in Figure 6-10. The severe deviations from the path in the 0.45s and 0.55s delay tests again reinforce the instabilities in these two conditions.

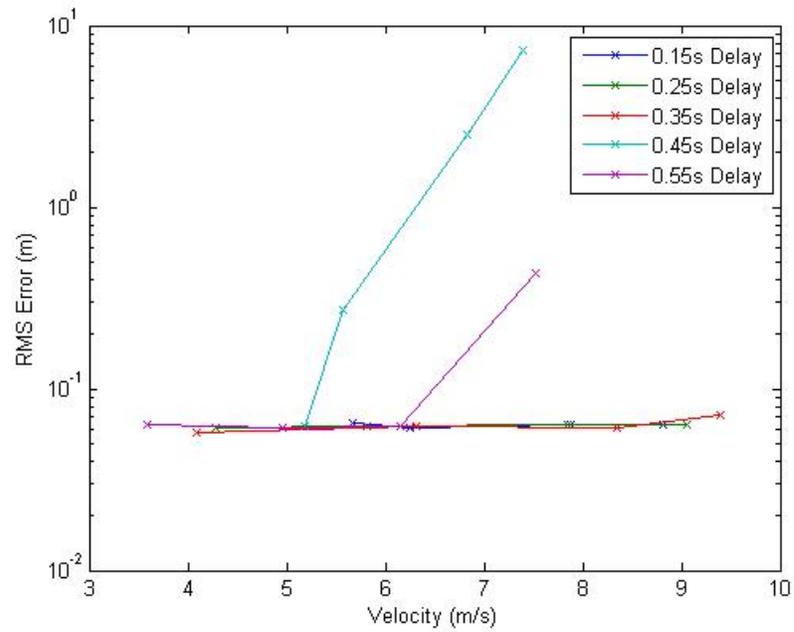


Figure 6-9. Root mean square error vs. velocity for various time delays

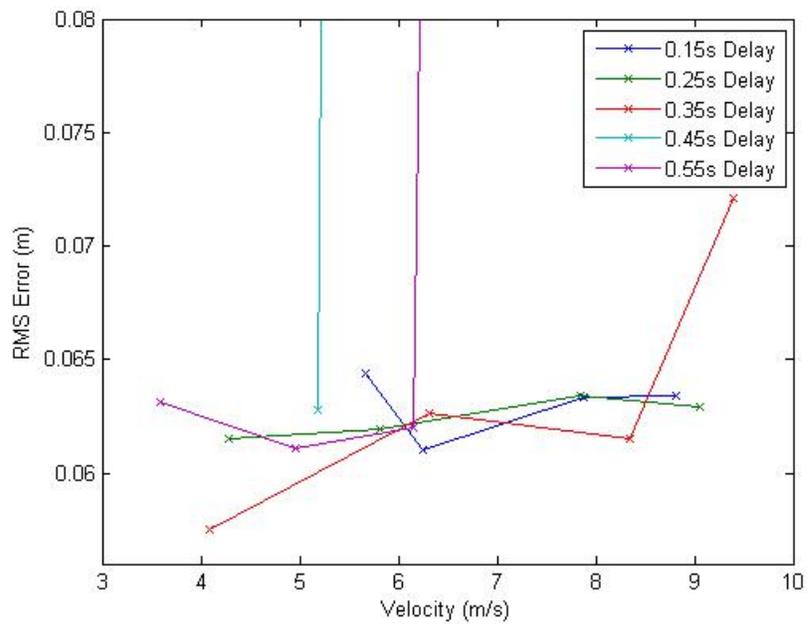


Figure 6-10. Zoomed plot of root mean square error vs. velocity for various time delays

Finally, the overall results for this test are shown in Table 6–2, including the full round-trip delay with the steering motor delay built in.

Table 6–2. Results from varying delay experiment

Test (One-way Delay)	Round-trip Delay w/ Motor Delay (s)	Speed (m/s)	RMS Error (m)
0s Delay	0.15	5.68	0.0644
	0.15	6.26	0.061
	0.15	7.89	0.0633
	0.15	8.82	0.0634
0.05s Delay	0.25	4.29	0.0615
	0.25	5.82	0.0619
	0.25	7.85	0.0634
	0.25	9.05	0.0629
0.1s Delay	0.35	4.09	0.0575
	0.35	6.31	0.0626
	0.35	8.35	0.0615
	0.35	9.39	0.0721
0.15s Delay	0.45	5.19	0.0628
	0.45	5.57	0.2740
	0.45	6.83	2.5193
	0.45	7.40	7.3842
0.2s Delay	0.55	3.59	0.0631
	0.55	4.96	0.0611
	0.55	6.15	0.0620
	0.55	7.53	0.4315

6.2.3 Discussion

The results from this experiment mostly agree with the simulated results. The simulation showed that the vehicle should enter instability between 0.35s and 0.4s of delay at 10m/s. In real tests the vehicle began to enter instability between 0.45s and 0.55s of delay by 7m/s. While these delays are slightly higher than predicted by the simulation, the speed at which they were reached

is slightly lower. To view the validity of these results, the simulation was run under the parameters of the physical test. As predicted by the physical results, the vehicle is unstable at 7.5m/s with 0.55s delay and 7.4m/s with 0.45s delay in the simulation, as shown in Figure 6-11 and Figure 6-12. However, as shown in Figure 6-13, at 6.8m/s with 0.45s delays the vehicle is not predicted to go unstable, but experience heavy oscillations before settling. This suggests that humans may use a preview time shorter than 0.5s. However, by decreasing the preview time by only 0.07s to 0.43s, the vehicle will go unstable at 6/8ms and 0.45s delay. This is shown in Figure 6-14. Overall, this result shows that the MacAdam model at 0.5s preview time provides an accurate model of the vehicle stability given certain system delays.

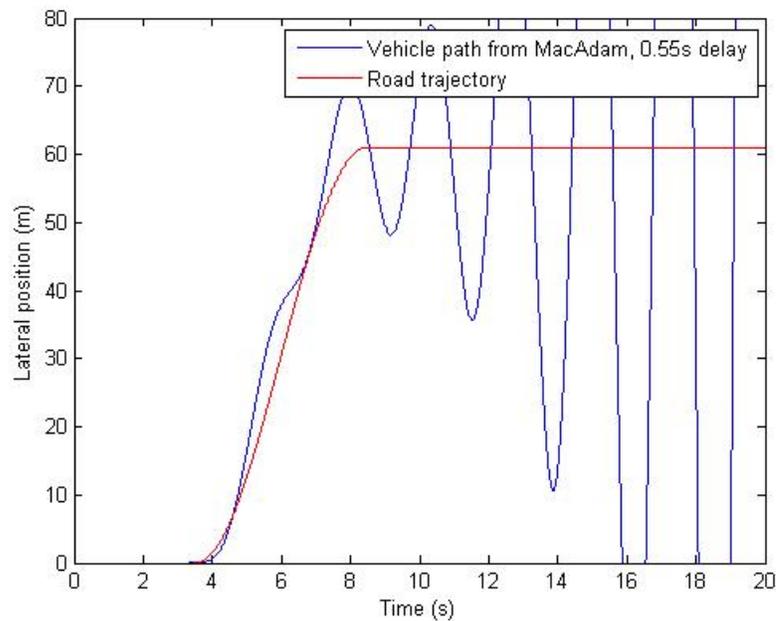


Figure 6-11. Lane change maneuver at 7.53m/s with 0.55s delay

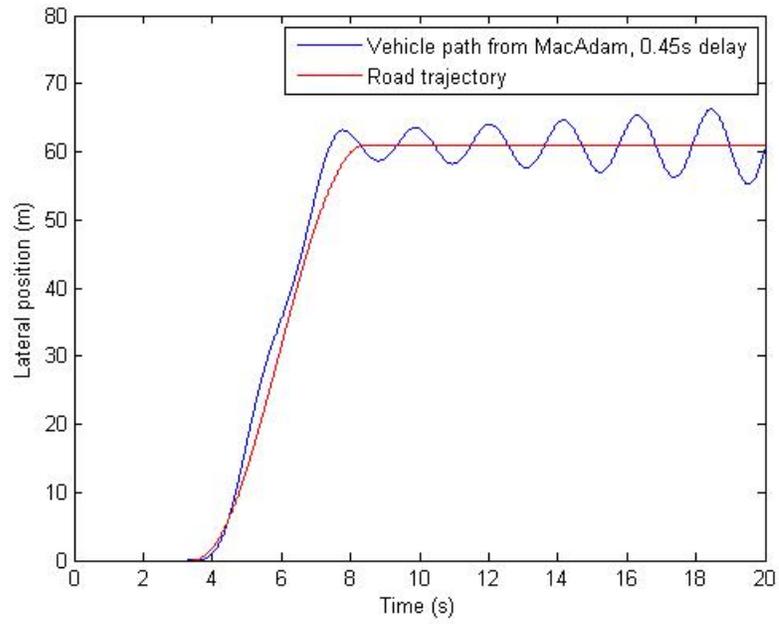


Figure 6-12. Lane change maneuver at 7.4m/s with 0.45s delay

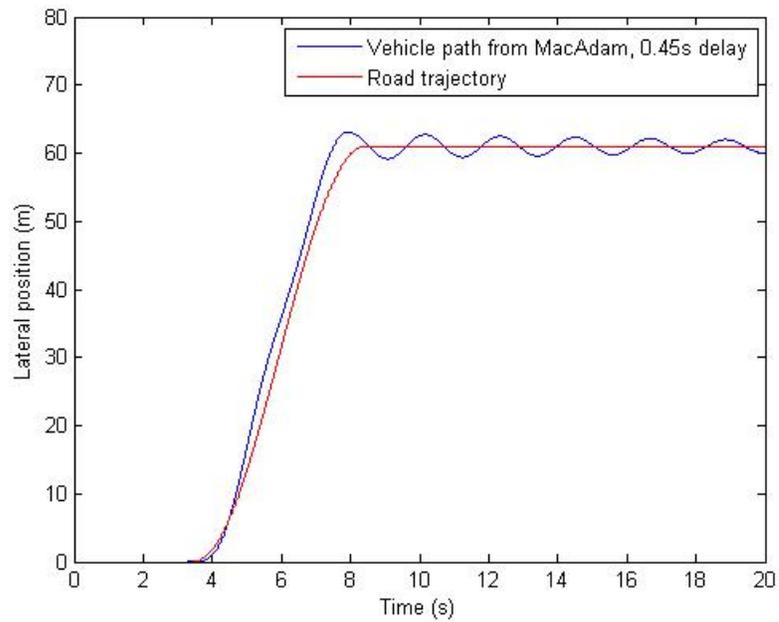


Figure 6-13. Lane change maneuver at 6.8m/s with 0.45s delay

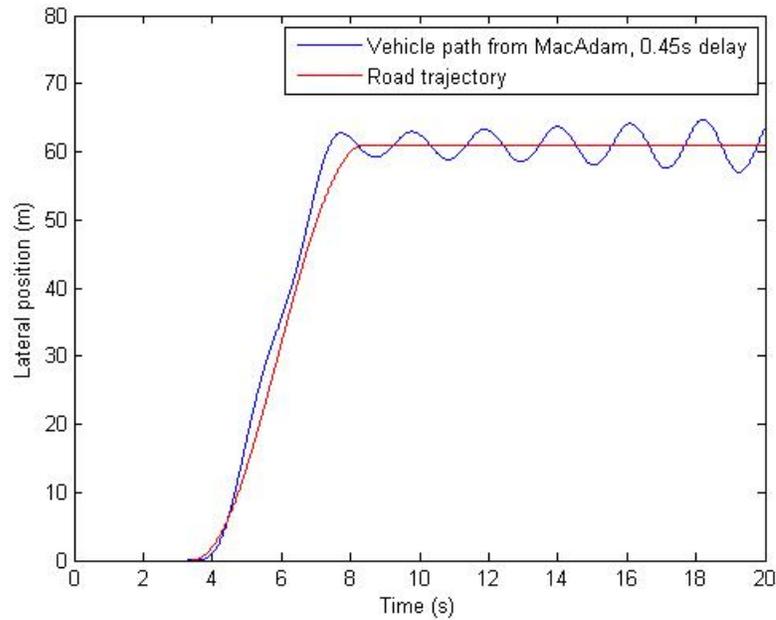


Figure 6-14. Lane change maneuver at 6.8m/s with 0.45s delay, but at 0.43s preview

The results also point to the possibility that humans can handle larger delays than the simulation. The reason that a human driver can withstand more delay is likely due to the fact that the human driver is able to use a variable level of preview time while the simulation is set to one specific preview time. For example, if the system experiences an abnormally large delay then the human can just look further away to adapt to the situation. However, with the simulation, if the delay is large, the preview time cannot be adapted to the situation and regain control. This supports the view that humans naturally drive with very short preview time (under 0.5s) for basic lane keeping, but use longer distances for more advanced planning.

The result that instability was reached earlier in the 0.45s test than in the 0.55s test is interesting. While it suggests that the vehicle is more unstable with a 0.45s delay it is not conclusive. The test subject requesting that speed be maintained during the instabilities of a 0.45s delay shows that, while the vehicle was departing from the path, it was not fully unstable and perhaps simply oscillatory. This is held in contrast to the 0.55s delay test where the human driver

immediately requested that the speed be decreased upon losing control of the vehicle. The reason that the vehicle went unstable in the 0.45s delay test before the 0.55s delay test is likely that the vehicle was on the border of stable and oscillatory state and either a human factor or abnormally long delay occurred to push the system across that border.

Overall, this experiment indicates that drivers can very easily enter an unstable state at delays comparable to the ones found in the simulation, but a human may actually be able to withstand slightly higher delays by using the capacity for variable preview times over the simulation's set single-point preview.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis presented the goal of determining the specific feedback cues used by humans when driving a vehicle. It is desirable to determine which feedback cues are most important for the implementation of teleoperated systems because, while full immersion into a remote environment is both highly challenging and expensive, a significant level of immersion can be achieved by focusing on the primary feedback cues. Specifically in this work, the primary questions were where is the human driver looking when operating a vehicle, and how much preview time does a human driver need to safely control a vehicle? By knowing the answer to these questions, the effect of delays on the teleoperated system can be managed more safely and efficiently.

Testing conditions were set for both simulated and real tests. In order to provide an accurate simulation, both the test vehicle and human test subject were modeled. The test vehicle was modeled using the linear bicycle model, augmented with a linear fit to the non-linear response of the teleoperated steering motor. The human driver was modeled using the MacAdam controller that estimates the displacement of the vehicle from a single point on a path trajectory at some time in the future (preview time) based on a constant speed and steering angle, and uses that displacement to maintain close path tracking.

In Test 1, the simulated response of the vehicle given no delays and variable preview times was measured. This simulation showed that a driver can maintain stability with small preview times ($< 0.2s$). This proved difficult to test in practice at safe testing speeds, and thus could not be validated in the testing environment provided. More importantly, however, the

simulation provided an estimated relationship between phase margin, preview distance, and the respective delay time that can be tolerated with each phase margin.

In Test 2, the simulated response of the vehicle given a set preview and varying delays in the feedback and steering command was measured. The preview was chosen to be 0.5s, which was the most realistic preview time based on Test 3's results. The simulation showed that drivers can easily reach instability in vehicles when given only a 0.31s delay at 10m/s. This result was then compared to those from Test 1 to find that the phase margin of a human driver is about 60° , which also fits with the lane change response plots that show the driver is a slightly under-damped system.

Test 3 measured the real-world response of the vehicle during teleoperation given no delays and variable preview time. The results from this test support those from Test 1 because the vehicle never reached a point of instability. It is, however, very interesting to note that while Test 1 predicted closer path tracking with less preview time, this test showed fairly constant tracking across all preview times. This result leads to the belief that, while humans are normally presented a large amount of preview, they actually only use what is very close for basic path tracking. In fact, Test 2 suggests that humans only need about 0.35s of preview to drive a vehicle in a stable condition. For most vehicles at safe speeds, this preview time equates to a distance that is actually under the vehicle, suggesting that humans may drive by extrapolating visual data into invisible areas and using that to guide the vehicle.

Finally, Test 4 measured the real-world response of the vehicle during teleoperation given unrestricted preview time and variable delays. The results support those from Test 2, showing a similar range in which instabilities will occur. Test 2 predicted that instabilities occur around 0.35s to 0.4s of delay at 10m/s, for 0.5 second preview, while Test 4 shows that the instabilities arose at longer delays of 0.45s to 0.55s, but at a lower speed of only about 7m/s. By running additional simulations the use of 0.5s preview time was confirmed as accurate, and the

MacAdam controller was shown to be an accurate representation of vehicle stability with transmission delays.

These numbers have large uncertainty, but still are extremely important when determining the safety and performance factors in teleoperated vehicles. It is also important to note how small the delays are that result in instability, even at low speeds. The results suggest that values on the order of 0.3 seconds are sufficient to destabilize a remotely-operated vehicle. It is clear that, in order to make a safe, high-performance teleoperated vehicle there must be both a rugged system in place and an architecture by which delays in the system are detected and the vehicle is controlled.

Most importantly, this work gives a base of evidence that shows where a driver is looking when operating a vehicle. From the evidence presented, it is likely that a human is visually inspecting the environment at the closest point of reference, and it is also likely that he/she is inferring the location of that point of reference to only about 0.35s in front of the vehicle's center of gravity. This is important because it provides defense for the use of various driver models when running simulations. For example, the MacAdam model can be run at any preview time and, if a large preview time is chosen, the simulation will show very poor path tracking ability. This work can be used as evidence as to how preview should be used when modeling human drivers.

7.2 Future Work

Like with most scientific works, this thesis creates more questions than it answers, which provides the opportunity for many follow-up experiments. These could include any goals from testing the instability point of limited preview driving to finding other non-visual feedback cues used by the driver.

In order to test the instability point of limited preview driving, tests must be run at very high speeds. In order to perform these tests safely, path tracking should be performed on a straight line while small perturbations are added to the steering input. The test subject would then be required to compensate for those perturbations through their own steering commands. A very open testing area like an airport runway would be required for this test. Or, one could perform these experiments using only a simulated environment.

This thesis infers that humans use close preview points to guide the vehicle even when presented with longer distance points of reference. This can be tested by providing the human driver view of only that specific preview point instead of limiting the view to a specific preview point and below. In an expansion to this test, this would be accomplished by placing a black bar at the top and bottom of the screen. To truly exclude all other visual cues, a blank screen showing only the point of reference could be presented to the driver while he/she is instructed to simply follow the line. Further, the line being tracked could be varied (or a disturbance added) so that the geometry of the curve is not predictable.

The accuracy of the simulations themselves could be verified by testing the real vehicle, but having the steering controlled by the MacAdam controller as performed by Brown [56]. This test would provide a nice comparison between a simulated test versus real-world test.

Non-visual cues and risk homeostasis effects can be tested by completely removing the human driver from the vehicle and repeating Test 3 and Test 4. In the MacAdam model and the physical tests run in this work, inertial feedback (the movement of the vehicle itself) is used to aid in the path planning. While the inertial feedback cannot be decoupled from the MacAdam model, the removal of the human driver from the vehicle does decouple the inertial feedback from the physical test. It is beneficial to determine the effect of inertial feedback when considering the use of driving simulators as an instrument for teleoperation. If inertial feedback is used heavily by the

human driver, the use of a motion base enabled driving simulator should greatly increase the performance of the teleoperated vehicle.

This method could be used for modeling systems with different dynamics. Through both simulation and physical testing, anything from skid steer ground robots to boats could be analyzed. It would be interesting to determine if similar phase or preview time is used when guiding these systems.

Finally, it would be beneficial to repeat these tests numerous times using multiple test subjects. While the time required and difficulty of running a single test did not allow the repeated testing of multiple subjects in the time allotted for this work, having more points of data would further solidify the data presented here.

REFERENCES

- 1] G. Niemeyer, C. Preusche and G. Hirzinger, "Telerobotics," in *Springer Handbook of Robotics*, Springer, 2008, pp. 741-757.
- 2] C. Preusche and G. Hirzinger, "Haptics in Telerobotics," *The Visual Computer*, vol. 23, no. 4, pp. 273-284, 2007.
- 3] T. Fong and C. Thorpe, "Vehicle Teleoperation Interfaces," *Autonomous Robots*, vol. 11, pp. 9-18, 2001.
- 4] A. Kelly, N. Chan, H. Herman, R. Meyers, P. Rander, R. Warner and J. Ziglar, "Real-time photorealistic virtualized reality interface for remote mobile robot control," *The International Journal of Robotics Research*, vol. 30, pp. 384-404, 2010.
- 5] P. Hancock and T. Sheridan, "The future of driving simulators," in *Handbook of Driving Simulation for Engineering Medicine*, Boca Raton, FL, CRC Press, 2011, pp. 4-1-4-11.
- 6] P. McNamara, "Inside Toyota's \$15m driving simulator," 26 February 2013. [Online]. Available: <http://www.carmagazine.co.uk/Community/Car-Magazines-Blogs/Phil-McNamara-Blog/Inside-Toyotas-15m-driving-simulator/>. [Accessed 26 February 2013].
- 7] J. Olstam and S. Espié, "Combination of autonomous and controller vehicles in driving simulator scenarios," in *International Conference of Road Safety and Simulation*, 2007.
- 8] G. Wilde, "Risk homeostasis theory: An overview," *Injury Prevention*, vol. 4, pp. 1-6, 2007.
- 9] S. Wright, N. Ward and A. Cohn, "Enhanced presence in driving simulators using autonomous traffic with virtual personalities," *Presence*, vol. 11, pp. 578-590, 2002.
- 10] S. Brennan, "Modeling and control issues associated with scaled vehicles," M.S. thesis, Dept. Mech. Eng., University of Illinois at Urbana-Champaign, IL, 1999.
- 11] S. Brennan, "On size and control: The use of dimensional analysis in controller design," Ph.D. dissertation, Dept. Mech. Eng., University of Illinois, Urbana-Champaign, IL, 2002.
- 12] S. Lapamong, "Vehicle similtude modeling and validation of the Pennsylvania State University rolling roadway simulator," M.S. thesis, The Pennsylvania State Univeristy, University Park, 2007.
- 13] P. Onesti, "Combination of Human- and Hardware-in-the-Loop Testing Reduces Development Time," *ATZautotechnology*, vol. 9, pp. 32-37, 2009.
- 14] H. Fathy, Z. Filipi, J. Hagen and J. Stein, "Review of hardware-in-the-loop simulation and its prospects in the automotive area," in *Proceedings of SPIE*, 2006.
- 15] R. Aracil, M. Buss, S. Cobos, M. Ferre, S. Hirche, M. Kuschel and A. Peer, "The Human Role in Telerobotics," in *Advances in Telerobotics*, Berlin, Springer, 2007, pp. 11-24.
- 16] S. Hirche, M. Ferre, J. Barrio, C. Melchiorri and M. Buss, "Bilateral Control Architectures for Telerobotics," *Advances in Telerobotics*, vol. 31, pp. 163-176, 2007.
- K. W. Ong, G. Seet and S. K. Sim, "An Implementation of Seamless Human-Robot

- 17] Interaction for Telerobotics," *International Journal of Advanced Robotic Systems*, vol. 5, no. 2, pp. 167-176, 2008.
- I. Manuaba, K. Taylor and T. Gedeon, "Evaluation of Response Models to a Series of Commands in a Telerobotics Supervisory Control System," in *ICIRA*, Berlin, Springer, 18] 2012, pp. 73-84.
- J. Bruder, D. Klimentjew and J. Zhang, "Haptic Interactive Telerobotics Control Architecture With Iterative Solving Of Inverse Kinematics," in *IEEE International Conference on Robotics and Biomimetics*, Phuket, Thailand, 2011.
- 19] T. Fong, F. Conti, S. Grange and C. Baur, "Novel interfaces for remote driving: gesture, haptic and PDA," *Mobile Robots XV and Telepresence Technologies VII*, vol. 4195, pp. 300-311, 2001.
- 20] D. Toffin, G. Reymond, A. Kemeny and J. Droulez, "Role of steering wheel feedback on driver performance: driving simulator and modeling analysis," *Vehicle System Dynamics*, vol. 45, no. 4, pp. 375-388, 2007.
- 21] N. Tanner and G. Niemeyer, "Improving Perception in Time-delayed Telerobotics," *The International Journal of Robotics Research*, vol. 24, no. 8, pp. 631-644, 2005.
- 22] S. Shoval, M. Wagner and E. Porat, "Reconstructing Haptic Sensing in Tele-Robotics Surgery by Non-Attentive Visual Method," in *International Conference on Robotics and Biomimetics*, Phuket, Thailand, 2011.
- 23] T. L. Brooks, "Telerobotic response requirements," in *Man and Cybernetics*, 1990.
- 24] M. A. Al-Mouhamed, M. Nazeeruddin and N. Merah, "Design and Instrumentation of Force Feedback in Telerobotics," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1949-1957, 2009.
- 25] M. Chouiten and C. Domingues, "Distributed Mixed Reality for remote underwater telerobotics exploration," in *Laval Virtual VRIC*, Laval, France, 2012.
- 26] M. Rozeff, "The LRC Blog," 3 February 2011. [Online]. Available: 27] <http://www.lewrockwell.com/blog/lewrw/archives/77218.html>. [Accessed 30 June 2013].
- 28] B. Nguyen and J. Ryu, "Haptic Interface for Intuitive Teleoperation of Wheeled and Tracked Vehicles," *International Journal of Automotive Technology*, vol. 13, no. 6, pp. 949-954, 2012.
- 29] B. Nguyen and J. Ryu, "Design of a Master Device for the Teleoperation of Wheeled and Tracked Vehicles," in *International Conference on Control, Automation, and Systems*, Gyeonggi-do, Korea, 2010.
- 30] T. Ersal, M. Brudnak, A. Salvi, J. Stein, Z. Filipi and H. Fathy, "Development and model-based transparency analysis of an internet-distributed hardware-in-the-loop simulation platform," *Mechatronics*, vol. 21, no. 1, pp. 22-29, 2011.
- 31] T. Ersal, B. Gillespie, M. Brudnak, J. Stein and H. Fathy, "Effect of Coupling Point Selection on Distortion in Internet-distributed Hardware-in-the-Loop Simulation," in

- American Control Conference*, San Francisco, CA, 2011.
- 32] A. Halme, J. Suomela and M. Savela, "Applying telepresence and augmented reality to teleoperate field robots," *Robotics and Autonomous Systems*, vol. 26, pp. 117-125, 1999.
- 33] O. Goldstain, I. Ben-Gal and Y. Bukchin, "Evaluation of Telerobotic Interface Components for Teaching Robot Operation," *IEEE Transactions on Learning Technologies*, vol. 4, no. 4, pp. 365-376, 2011.
- 34] F. Tanaka and T. Takahashi, "Linking Children by Telerobotics: Experimental Field and the First Target," in *HRI '11*, Lausanne, Switzerland, 2011, pp. 267-268.
- 35] L. Brayda, N. Mollet and R. Chellali, "Mixing Telerobotics and Virtual Reality for Improving Immersion in Artwork Perception," in *Edutainment*, Berlin, Springer, 2009, pp. 63-73.
- 36] R. Marin, P. J. Sanz and J. S. Sanchez, "A Very High Level Interface to Teleoperate a Robot via Web," in *International Conference on Robotics & Automation*, Washington, DC, 2002.
- 37] F. Moutaouakkil, M. El Bakkali and H. Medromi, "A New Approach of Telerobotic over Internet," in *World Congress on Engineering and Computer Science*, San Francisco, CA, 2010.
- 38] H. H. King, B. Hannaford, K. Kwok, G. Yang, P. Griffiths, A. Okamura, I. Farkhatdinov, J. Ryu, G. Sankaranarayanan, V. Arikatla, K. Tadano, K. Kawashima, A. Peer, T. Schaub, M. Buss, L. Miller, D. Glozman, J. Rosen and R. Low, "Plugfest 2009: Global Interoperability in Telerobotics and Telemedicine," in *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010.
- 39] S. Grange, T. Fong and C. Baur, "Effective Vehicle Teleoperation on the World Wide Web," in *International Conference on Robotics and Automation*, San Francisco, CA, 2000.
- 40] Encyclopedia Britannica, "Wi-Fi," 30 June 2013. [Online]. Available: <http://www.britannica.com/EBchecked/topic/1473553/Wi-Fi>. [Accessed 30 June 2013].
- 41] M. Michalak, "Multimedia transmission in Bluetooth-based networks," M.S. Thesis, University of Mining and Metallurgy, Cracow, Poland, 2001.
- 42] R. Wolff, "Intro to DSRC," 30 September 2003. [Online]. Available: <http://www.coe.montana.edu/ee/rwolff/shel%20leader%20dsrc.pdf>. [Accessed 30 June 2013].
- 43] P. DeBeasi, "802.11n: The End of Ethernet?," 14 September 2009. [Online]. Available: http://www.arubanetworks.com/pdf/technology/whitepapers/wp_Burton_End_of_Ethernet.pdf. [Accessed 6 July 2013].
- 44] "Bluetooth Modules FAQ," Free2Move, 6 July 2013. [Online]. Available: http://www.free2move.se/?page_id=1058. [Accessed 6 July 2013].
- 45] J. Guo, "Vehicle Safety Communications in DSRC," 2006. [Online]. Available: http://groups.engin.umd.umich.edu/vi/w5_workshops/guo_DSRC.pdf. [Accessed 7 July 2013].
- N. J. Ploplys, P. A. Kawka and A. G. Alleyne, "Closed-Loop Control over Wireless

- 46] Networks," *IEEE Control Systems Magazine*, pp. 58-71, June 2004.
P. A. Kawka and A. G. Alleyne, "An Analysis Framework for Evaluating Dropout
47] Compensation Strategies in Wireless Servo Systems," *Journal of Dynamic Systems,
Measurement, and Control*, vol. 130, 2008.
P. A. Kawka and A. G. Alleyne, "Stability and Performance of Packet-Based Feedback
48] Control over a Markov Channel," in *American Control Conference*, Minneapolis, MN, 2006.
I. Elhajj, N. Xi, W. K. Fung, Y. Liu, Y. Hasegawa and T. Fukuda, "Supermedia-Enhanced
49] Internet-Based Telerobotics," *Proceedings of IEEE*, vol. 91, no. 3, pp. 396-421, 2003.
S. F. Atashzar, H. A. Talebi, M. Shahbazi, F. Towhidkhah and R. V. Patel, "Time Delayed
50] Non-minimum Phase Slave Tele-Robotics," in *IEEE Conference on Decision and Control*,
Orlando, FL, 2011.
G. Wen, M. Gong and H. Zhang, "Research on a Development Hydraulic Servo Tele-
51] robotics System," in *International Conference on Mechatronics and Automation*, Beijing,
China, 2011.
J. Harder and U. Walter, "Communication Architecture Evaluation for Real-Time Tele-
52] Operated Spacecraft," in *IEEE Aerospace Conference*, Big Sky, MT, 2012.
B. Khiter, A. Hentout, E. Boutellaa, M. Benbouali and B. Bouzouia, "Internet-Based
53] Telerobotics of Mobile Manipulators: Application on RobuTER/ULM," in *ICIRA*, Berlin,
Springer, 2012, pp. 635-644.
"ROS Wiki," 31 May 2013. [Online]. Available: <http://www.ros.org/wiki/>. [Accessed 31
54] May 2013].
S. Lapamong, "Vehicle Rollover Prediction for Banked Surfaces," Ph.D dissertation, Dept.
55] Mech. Eng., Penn State University, University Park, PA, 2010.
A. Brown, "Model-based Inclusion of Previewed Information for Lateral Vehicle State
56] Estimation," Ph.D. dissertation, Dept. Mech. Eng., Penn State University, University Park
PA, 2013.
C. MacAdam, "Application of an Optimal Preview Control for Simulation of Closed-Loop
57] Automobile Driving," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6,
pp. 393-399, 1981.
P. Stankiewicz, "Vehicle Path Following and Rollover Prevention Using Previewed State
58] Information," B.S. thesis, Dept. Mech. Eng., Penn State University, University Park, PA,
2013.
A. Ungoren and H. Peng, "An adaptive lateral preview driver model," *Vehicle System
59] Dynamics*, vol. 43, no. 4, pp. 245-256, 2006.

APPENDIX A – DERIVATION OF THE BICYCLE MODEL WITH THE MOTOR MODEL

The standard bicycle model is given by Equation A-1 and the state space form of the steering motor transfer function is given by Equation A-2.

$$\begin{bmatrix} \dot{y} \\ \dot{V} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & U \\ 0 & \frac{C_{\alpha f} + C_{\alpha r}}{mU} & \frac{aC_{\alpha f} - bC_{\alpha r}}{mU} - U & 0 \\ 0 & \frac{aC_{\alpha f} - bC_{\alpha r}}{I_z U} & \frac{a^2 C_{\alpha f} + b^2 C_{\alpha r}}{I_z U} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ V \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{C_{\alpha f}}{m} \\ -\frac{aC_{\alpha f}}{I_z} \\ 0 \end{bmatrix} \delta_f \quad \text{Equation A-1}$$

$$y = [1 \quad 0 \quad 0 \quad 0]$$

$$\dot{x} = \begin{bmatrix} -161 & -7215 & -122500 & -1100000 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_{comm}$$

Equation A-2

$$\delta_f = [0 \quad 0 \quad -11000 \quad 1100000] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Since the inputs to the two state space models are different it is challenging to make a single augmented matrix. However, the input for Equation A-1 is the output of Equation A-2, so the relationship shown in Equation A-3 is calculated.

$$B_1 * \delta_f = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 11000 \left(\frac{C_{\alpha f}}{m} \right) & -1100000 \left(\frac{C_{\alpha f}}{m} \right) \\ 0 & 0 & 11000 \left(\frac{aC_{\alpha f}}{I_z} \right) & -1100000 \left(\frac{aC_{\alpha f}}{I_z} \right) \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{Equation A-3}$$

Using this matrix, the full state space model from steering command δ_{comm} to vehicle states is calculated in Equation A-4 through Equation A-8.

$$A_m = \begin{bmatrix} 0 & 1 & 0 & U & 0 & 0 & 0 & 0 \\ 0 & \frac{C_{\alpha f} + C_{\alpha r}}{mU} & \frac{aC_{\alpha f} - bC_{\alpha r}}{mU} - U & 0 & 0 & 0 & 11000 \left(\frac{C_{\alpha f}}{m} \right) & -1100000 \left(\frac{C_{\alpha f}}{m} \right) \\ 0 & \frac{aC_{\alpha f} - bC_{\alpha r}}{I_z U} & \frac{a^2 C_{\alpha f} + b^2 C_{\alpha r}}{I_z U} & 0 & 0 & 0 & 11000 \left(\frac{aC_{\alpha f}}{I_z} \right) & -1100000 \left(\frac{aC_{\alpha f}}{I_z} \right) \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -161 & -7215 & -122500 & -1100000 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{Equation A-4}$$

$$B_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Equation A-5

$$C_m = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -11000 \quad 1100000]$$

Equation A-6

$$D_m = 0$$

Equation A-7

$$\dot{x} = A_m x + B_m \delta_{comm}$$

Equation A-8

$$y = C_m x + D_m \delta_{comm}$$

APPENDIX B – DERIVATION OF THE FEEDBACK EQUATIONS FOR THE MACADAM CONTROLLER

The feedback equation with the motor model is calculated first. The error between the output y and input δ_{comm} is shown in Equation B-1, which is then plugged into the state space equation to get Equation B-3.

$$\delta = K^*(y_{in} - (C_m x + D_m \delta_{comm})) \quad \text{Equation B-1}$$

$$\dot{x} = A_m x - B_m K^*(C_m x + D_m \delta_{comm}) + B_m K^* y_{in} \quad \text{Equation B-2}$$

$$\dot{x} = (A_m - B_m K^* C_m)x - B_m K^* D_m \delta_{comm} + B_m K^* y_{in} \quad \text{Equation B-3}$$

Finally, knowing Equation B-4 and using the above derivation, the final feedback equation for the system with the motor model is calculated in Equation B-6.

$$\delta_{comm} = C_m x \quad \text{Equation B-4}$$

$$\dot{x} = (A_m - B_m K^* C_m)x - B_m K^* D_m C_m x + B_m K^* y_{in} \quad \text{Equation B-5}$$

$$\dot{x} = (A_m - B_m K^* C_m - B_m K^* D_m C_m)x + B_m K^* y_{in} \quad \text{Equation B-6}$$

Next, the feedback equation without the motor model is calculated. Equation B-2 and Equation B-3 still hold true, but there is a different equation for δ shown in Equation B-7. Using this equation, the final equation for feedback without a motor model is found in Equation B-9

$$\delta = \frac{K^*(y_{in} - C_o x)}{1 + K^* D_o} \quad \text{Equation B-7}$$

$$\dot{x} = A_o x + B_o \left(\frac{K^*(y_{in} - C_o x)}{1 + K^* D_o} \right) \quad \text{Equation B-8}$$

$$\dot{x} = \left(A_o - \frac{B_o K^* C_o}{1 + K^* D_o} \right) x - \left(\frac{B_o K^*}{1 + K^* D_o} \right) y_{in} \quad \text{Equation B-9}$$