# DEVELOPMENT OF ELECTRICAL AND CONTROL SYSTEM OF AN UNMANNED GROUND VEHICLE FOR FORCE FEEDBACK TELEOPERATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AKİF HACINECİPOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

DEVELOPMENT OF ELECTRICAL AND CONTROL SYSTEM OF AN UNMANNED GROUND VEHICLE FOR FORCE FEEDBACK TELEOPERATION

submitted by **AKİF HACINECİPOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Suha Oral _____
Head of the Department, **Mechanical Engineering**

Assist. Prof. Dr. E. İlhan Konukseven _____
Supervisor, **Mechanical Engineering Dept., METU**

Assist. Prof. Dr. A. Buğra Koku _____
Co-Supervisor, **Mechanical Engineering Dept., METU**

**Examining Committee Members:**

Assist. Prof. Dr. Yiğit Yazıcıoğlu _____
Mechanical Engineering Dept., METU

Assist. Prof. Dr. E. İlhan Konukseven _____
Mechanical Engineering Dept., METU

Assist. Prof. Dr. A. Buğra Koku _____
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Erol Şahin _____
Computer Engineering Dept., METU

Assist. Prof. Dr. Afşar Saranlı _____
Electrical and Electronics Engineering Dept., METU

**Date:** September 14th, 2012

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name: Akif Hacınecipoğlu

Signature :

## ABSTRACT

DEVELOPMENT OF ELECTRICAL AND CONTROL SYSTEM OF AN UNMANNED GROUND VEHICLE FOR FORCE FEEDBACK TELEOPERATION

Hacınecipoğlu, Akif
M.Sc., Department of Mechanical Engineering
Supervisor : Assist. Prof. Dr. E. İlhan Konukseven
Co-Supervisor : Assist. Prof. Dr. A. Buğra Koku

September 2012, 156 pages

Teleoperation of an unmanned vehicle is a challenging task for human operators especially when the vehicle is out of line of sight. Improperly designed and applied display interfaces directly affect the operation performance negatively and even can result in catastrophic failures. If these teleoperation missions are human-critical then it becomes more important to improve the operator performance by decreasing workload, managing stress and improving situational awareness. This research aims to develop electrical and control system of an unmanned ground vehicle (UGV) using an All-Terrain Vehicle (ATV) and validate the development with investigation of the effects of force feedback devices on the teleoperation performance. After development, teleoperation tests are performed to verify that force feedback generated from the dynamic obstacle information of the environment improves teleoperation performance. Results confirm this statement and the developed UGV is verified for future research studies. Development of UGV, algorithms and real system tests are included in this thesis.

**Keywords:** Unmanned Ground Vehicle, Teleoperation, Obstacle Avoidance, Force Feedback, Human Robot Interaction

# ÖZ

BİR INSANSIZ KARA ARACININ KUVVET GERİ BESLEME DESTEKLİ UZAKTAN KONTROLÜ İÇİN ELEKTRİK VE KONTROL SİSTEMİNİN GELİŞTİRİLMESİ

Hacınecipoğlu, Akif
Yüksek Lisans, Makine Mühendisliği Bölümü
Tez yöneticisi : Yrd. Doç. Dr. E. İlhan Konukseven
Ortak tez yöneticisi : Yrd. Doç. Dr. A. Buğra Koku

Eylül 2012, 156 sayfa

İnsansız araçların uzaktan kontrolü, özellikle araç görüş alanı dışında ise kullanıcılar için oldukça zorlu bir görev haline gelmektedir. Hatalı olarak tasarlanmış ve uygulanmış kontrol ara yüzleri operasyon başarımını doğrudan olumsuz olarak etkilemekte ve hatta yıkıcı sonuçlar doğurabilmektedir. Özellikle bu görevler insanlar için tehlikeli sayılabilecek görevler ise iş yükünü azaltarak, görev stresini kontrol altına alarak ve durum farkındalığını arttırarak kullanıcının başarımını geliştirmek daha da önemli hale gelmektedir. Bu tez kapsamında bir insansız kara aracı (İKA) için elektrik ve kontrol sistemleri geliştirilmiş ve geliştirilen aracın doğrulanması kuvvet geri beslemeli uzaktan kontrol yönteminin kullanıcı başarımına etkilerinin araştırılması ile sağlanmıştır. Aracın üzerindeki algılayıcılardan alınan engel bilgisi ile kullanıcının yönlendirilmesinin başarıma olumlu etki yapıp yapmadığı gözlenmiştir. Sonuçlar aracın gelecekteki araştırmalar için kullanılabileceğini ve kuvvet geri besleme aygıtlarının uzaktan kontrol yöntemi olarak kullanılmasının kullanıcının başarımında olumlu etki yaptığını göstermiştir.

**Anahtar Kelimeler:** İnsansız Kara Aracı, Uzaktan Kontrol, Engelden Kaçınma, Kuvvet Geri Besleme, İnsan Robot Etkileşimi

***To My Lovely Family***

## ACKNOWLEDGEMENTS

First of all, I would like to express my sincere appreciation to my supervisors Asst. Prof. Dr. E. İlhan Konukseven and Asst. Prof. Dr. A. Buğra Koku for their invaluable guidance, advice, criticism and support that made this study possible.

I would like to express my thanks to my dear friends and colleagues Hasan Ölmez, Burak Şamil Özden and Kamil Özden for their friendship and technical support throughout the thesis period and to my dear friend Göker Ertunç for his help and courage. I would also like to thank to my research group colleagues Muhammet Kasım Gönüllü, Deniz Mutlu, Rasim Aşkın Dilan, Kadri Buğra Özütemiz, Serkan Tarçın and Matin Ghaziani for those endless talks and ideas on the projects. Another appreciation goes to Nejat Ulusal for his support in developing the vehicle body. I want to thank to my friend Aykut Tamer for the good times we had and his moral support throughout the study.

Finally and most importantly, I would like to thank to my lovely family including my love and intended wife Fatmanur for their understanding, help, encouragement and faith in me.

## TABLE OF CONTENTS

## LIST OF TABLES

TABLES

## LIST OF FIGURES

FIGURES

## LIST OF SYMBOLS

SYMBOLS

| | |
|---|---|
| $R_T$ | Turning radius |
| $a$ | Wheelbase |
| $b$ | Distance between steering pivot axes |
| $\emptyset$ | Maximum inner steering angle |
| $e$ | Scrub radius |
| $u(t)$ | Controller output |
| $K_p$ | Proportional gain |
| $K_d$ | Derivative gain |
| $e(t)$ | Error |
| $K_i$ | Integral gain |
| $\tau$ | Integration variable |
| $\delta_f$ | Front steering angle |
| $\delta_r$ | Rear steering angle |
| $\Psi$ | Heading angle |
| $\beta$ | Slip angle |
| $m$ | Mass |
| $V$ | Velocity |
| $R$ | Radius of curvature |
| $v_x$ | Longitudinal velocity |
| $v_y$ | Lateral velocity |
| $v_\theta$ | Rotational velocity |

## LIST OF ABBREVIATIONS

ABBREVIATIONS

| | |
|---|---|
| AAAI | American Association of Artificial Intelligence |
| ARL | Army Research Laboratory |
| ATV | All-Terrain Vehicle |
| CEP | Circular Error Probability |
| DAQ | Data Acquisition |
| DARPA | The Defense Advanced Research Projects Agency |
| DGPS | Differential Global Positioning System |
| DWA | Dynamic Window Approach |
| EKF | Extended Kalman Filter |
| FCS | Future Combat Systems |
| FOV | Field of View |
| FPB | Force Protection Battlelab |
| FR | Frame Rate |
| GPS | Global Positioning System |
| HCI | Human Computer Interaction |
| HRI | Human Robot Interaction |
| IMU | Inertial Measurement Unit |
| INS | Inertial Navigation System |
| LADAR | Laser Radar |
| LIDAR | Light Detection and Ranging |
| LOA | Level of Autonomy |
| MRDS | Microsoft Robotics Developer Studio |
| MRT | Multiple Resource Theory |
| MS | Monoscopic |
| NAHRS | Navigation Attitude Heading Reference System |
| NASA | National Aeronautics and Space Administration |

| | |
|---|---|
| RCTA | Robotics Collaborative Technology Alliances |
| ROS | Robot Operating System |
| RTWT | Real Time Windows Target |
| SA | Situational Awareness |
| SBPL | Search-Based Planning Library |
| SD | Standard Deviation |
| SLAM | Simultaneous Localization and Mapping |
| SS | Stereoscopic |
| TLX | Task Load Index |
| UAV | Unmanned Aerial Vehicle |
| UGV | Unmanned Ground Vehicle |
| USAF | United States Air Force |
| USAR | Urban Search and Rescue |
| VFH | Vector Field Histogram |
| WAAS | Wide Area Augmentation System |
| WTC | World Trade Center |

# CHAPTER 1

# INTRODUCTION

Robotic technology is a growing field which has applications in many areas such as national security, entertainment, search and rescue, earth and space exploration, tactics operations, production, health care and personal assistance.

## 1.1 Recent History of Robotics

In early 70s robotic devices began to be used in structured environments like production and assembly lines in factories. In these lines robots were confined to move in a certain environment and do necessary movements at desired time. Because of these reasons there was no need to human interference under normal working conditions.

The need for robotic applications has grown throughout past years. People started to include robotic agents in their daily life for hard and/or time consuming applications. As a result mobile robotics has emerged. Starting from early 90s up to recent years, robotic applications moved out from their predefined paths and structured environments to dynamic environments. These new environments have unpredictable moving objects around and traversable areas are not well defined. Therefore a robotic agent should have to sense an object and take precautions according to its state. Mobile robotics topic is a multi-disciplinary field that includes control engineering, cognitive science, mechanical engineering, computer science, even sociology and human psychology.

## 1.2 Civilian Use of Robots

Robots are being used in civilian area in many applications. Especially in space research, robotic assets are being widely used because these missions may be dangerous for human-beings and infeasible. American National Aeronautics and Space Administration (NASA) is one of the organizations that spend large budgets to space explorations. NASA sent two unmanned robots (Opportunity and Spirit) to Martian surface in 2004. Though the connection with Spirit lost in 2010, Opportunity still continues its mission on Mars. Another effort in civilian use of robots is organized by The Defense Advanced Research Projects Agency (DARPA) of United States; Grand Challenge. In Grand Challenge, the aim was to race 150 mile route across the Mojave Desert with autonomous unmanned vehicles. The prize of the competition was $2,000,000 in 2005. In 2007, the organization named as Urban Challenge and the teams are encouraged to develop unmanned vehicles that can move autonomously in urban environments.

Robots are also getting involved in search and rescue applications. Researchers are working to get robotic agents into the search and rescue fields. In accordance with this aim, American Association of Artificial Intelligence (AAAI) hosts RoboCup Robot Rescue Event. Competitors are encouraged to develop a better urban search and rescue (USAR) applications. After the attacks on the World Trade Center (WTC) on September 11, 2001, robots are much more involved in USAR missions. The rubble of WTC is searched for victims with robotic agents. Also these robots are used for medical supplies to victims and to help structural engineers working in the area.

Also important research budgets are used for developing robots to be used in entertainment field. Sony is one of the companies that invent to have more realistic and satisfying agents. Sony's dog-like robot AIBO is an example of these efforts. Well-known humanoid robot ASIMO of Honda is also another and the most evolved example in this field.

### 1.3 Military Use of Robots

United States Army has a Future Combat Systems (FCS) department. FCS is involved in the development of unmanned systems. U.S. Army Research Laboratory (ARL) Robotics Collaborative Technology Alliances (RCTA) described their capabilities as “enhancing Soldier physical security and survivability, improving situational awareness and understanding, and conducting reconnaissance, surveillance, targeting and acquisition missions in an era of rapidly evolving operational and technological challenges” [1].

FCS developed command and control (C2) program in coordination with DARPA and has conducted several experiments from 2001 to 2005. In these experiments, 90 minute long battle exercises were conducted. The personnel in C2 vehicles controlled UGV and UAV assets.

In 2003, another series of experiments conducted to get useful feedbacks from Soldiers controlling unmanned assets. In these experiments, an all-terrain vehicle, a small UAV, an unmanned vehicle called PackBot which is developed by iRobot Inc., and unmanned ground sensors are demonstrated. Soldiers suggested that the robotic agents are useful and contributes to the understanding of the battlefield.

### 1.4 Mobile Robots and HRI

Mobile robot applications range from fully human-controlled to autonomous agents. In fully teleoperated systems, human has the full control over the perception and movement of the agent. Human perceives and interprets the sensory information and decides on motor commands. Role of the robot is confined to do whatever the operator asks. In fully autonomous systems, human does not intervene to the progress of the robot. Even the need to operate is decided by the robot itself. Although the peak point is defined as fully autonomous robots, human role always will be persistent in mobile robotic applications. In this aspect a new topic arises: Human-Robot Interaction (HRI). HRI study field emerged from the need to understand the interaction between robotic systems and humans.

There should be a communication channel between human and robotic agent to be able to interact. Type of this interaction is defined by the distance between peers. Depending on the distance, human-robot interaction is separated into two categories:

• *Remote interaction*: Defined as the interaction type when the human and the robot are separated in terms of space and/or time.

• *Proximate interaction*: If two peers are in their line of sight, this type of interaction is defined as proximate interaction.

Proximate interaction requires the robot and human being in the same location or even in same room. Interaction between personal robotic assistants and humans require proximate interaction including physical, social or emotive aspects.

Remote interaction included in mobile robotics. If remote interaction with the robot requires mobility, this category is divided into two sub categories. Interaction with a mobile robot to change its location is referred as “teleoperation”. On the other side, if the mission is to change a location of a remote object, this interaction is named as “telemanipulation”.

### 1.5 Scope of the Thesis

In this thesis, electrical and control system of an unmanned ground vehicle is developed and the effects of force feedback on the teleoperation performance are studied on the developed UGV. For this purpose electrical hardware is implemented on a previously designed All-Terrain Vehicle (ATV) chassis, software developed and teleoperation tests are performed to verify the UGV development and that force feedback to the operator generated from the dynamic obstacle information improves teleoperation performance. UGV design, software, algorithms and real system tests are included in this thesis. Electrical design and implementation are also in the scope of the thesis.

## 1.6 Outline of the Thesis

This study is divided mainly two sections according to the content. First section is about development of the unmanned ground vehicle (UGV) that is made ready for teleoperation purposes. In the second part human-robot interaction and force feedback implementation is described.

First chapter is the introduction to the topic. Second chapter includes literature survey performed on these two sub-topics. Development of the UGV is represented in Chapter 3. A brief introduction to human-robot interaction, teleoperation and multimodal interfaces are described in Chapter 4. In Chapter 5, the developed UGV is validated via experiments with a force feedback application and obstacle avoidance implementation. Finally last section is dedicated to conclusions and possible future research on the topic that is studied.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 Introduction

In this chapter a survey on recent research and applications on similar unmanned systems are presented. The chapter is divided into two parts. In the first section related work in the literature on unmanned ground vehicle (UGV) design and construction is mentioned. Components, design criterion, sub-systems and software used are discussed. In the second part of this chapter, factors affecting human-robot interaction performance and research applications in the literature are presented. This part focused mostly on multimodal displays that are used in vehicle teleoperation.

## 2.2 Design of Unmanned Ground Vehicle

In the past couple of decades, universities, research companies and laboratories carried out studies related to design and construction of UGVs. Unmanned systems are generally developed for outdoor use. Especially military based research programs studied on developing mobile platforms that will operate in unstructured environments and mostly off-road routes. Because of these reasons ATV (All-Terrain Vehicle) based solutions became popular amongst these research studies.

### 2.2.1 Hardware

The hardware used in ATV-based research platforms includes environmental sensing equipment, localization equipment, electric motors and drivers, power units and controller computers. Features and performance of this hardware affects the experimental success of research platforms.

#### 2.2.1.1 Mobility

National Institute of Standards and Technology in conjunction with United States Army conducted applications using ATV-like vehicle in automatic target acquisition [2]. Murphy and Legowik integrated an inertial measurement unit (IMU) and differential global positioning system (GPS) on a military all-terrain vehicle. Vehicle was used to track previously recorded path using a pure pursuit algorithm. Also a laser scanner implemented to avoid obstacles on the path.

An autonomous robotic vehicle for tactical distributed surveillance is developed by Institute for Complex Engineered Systems of Carnegie Mellon University [3]. The purpose of the researchers in the CyberScout project was to develop a ground sensor platform that will contribute to the awareness and mobility of small military units. They have developed a robotic ATV (Figure 2-1) during the research to develop algorithms for multi-agent collaboration, efficient perception, sensor fusion, distributed command and control and task decomposition.

Figure 2-1: CyberATV, modified vehicle (left) and original vehicle (right)

CyberATV is built on a Polaris Sportsman 500 ATV (Figure 2-1). Throttle, steering and breaking functions are actuated by a computer. They have used a proportional directional control valve to actuate a hydraulic piston to control the steering action. A resistive linear potentiometer is implemented to provide feedback about the steering angle of the front wheels. For the braking system they used the hydraulic system of

the original ATV. A directional valve is used to control the brake cylinder resulting in only on/off braking. They did not implement any explicit feedback to the brake system. The ATV is originally equipped with a 4-stroke internal combustion engine which is regulated by a throttle plate. An R/C servomotor is connected to the throttle plate to control the throttle of the vehicle. Plate setup incorporates a torsional spring to cut off the throttle when power is turned off. A tachometer is put to the gear box to feedback the speed information to the computer system. As a computer system they used a PC/104 for low level processing for locomotion, and three networked PCs for high level processing which is required for planning, perception and communications. The CyberATV is equipped with DGPS, and stereo and mono vision systems for perception and localization.

Team ENSCO, sponsored by a privately owned engineering company ENSCO Inc, modified a Honda Rincon ATV (Figure 2-2) and participated in DARPA Grand Challenge in 2004 [4]. They have made modifications to the vehicle body to enclose sensor systems and suspensions. The steering system has been controlled by a brushless DC servo motor attached to the steering shaft. Original engine and transmission system is preserved. Two independent braking systems are used: hydraulic and cable. By this way, if one of the brakes fails, the other one assures braking. The gearshift is controlled using electrical relays. A servo motor has been attached to the throttle. The sensor system included in the vehicle consists of GPS, DGPS, Stereo camera, magnetic compass and LIDAR system.


Figure 2-2: ENSCO, original vehicle (left) and modified vehicle (right)

For data processing purposes, they have implemented three computers into the vehicle. A distributed processing scheme has been used. One computer handled the sensor data while second computer processed map and developed path logic. Third computer handled vehicle control and system feedback.

Another DARPA Grand Challenge participant who has developed their project on an ATV is Team Spirit of Las Vegas [5]. The base vehicle design was 2003 Honda 4X4 ATV (Figure 2-3). They have maintained the original footprint of the vehicle. It was powered by 649cc four strokes Honda engine. Steering control has been replaced with a geared DC motor driving chain and sprockets. The stock vehicle was equipped with a standard Honda automatic transmission system involving torque converter with a 3-speed drive. A high torque servo motor is attached to switch between forward, reverse and neutral positions. The brake system is also actuated with two high-torque servo motors one for front and the other for rear brakes. Throttle existing on the stock vehicle is also actuated with a high-torque servo motor.

Figure 2-3: The Spirit of Las Vegas

Sensor systems have been implemented on ATV to sense the environment. GPS, DGPS, 3-axis gyroscope and camera systems are attached for these purposes. For data processing tasks, two computer systems have been applied. One computer was responsible for vehicle control while the other one for navigation and video processing.

Another ATV based project is developed in University of Siegen, Germany [6]. It is based on Yamaha Kodiak 400, 4x4 all-terrain vehicle (Figure 2-4). A DC motor is implemented to actuate the steering mechanism. Position feedback of the steering shaft is obtained using a resolver. Gear shifting, gas throttle, and brake system are all activated by magnetic valves. To sense the environment of the vehicle, ultrasonic sensors, Doppler radar, electronic compass are mounted on the vehicle. Also for obstacle detection purposes a tilting line laser scanner has been attached to the front of the UGV.

Figure 2-4: Amor, original vehicle (left) and modified vehicle (right)

United States Air Force (USAF) Force Protection Battlelab (FPB) has developed another ATV-based project named Redcar Scout [7] (Figure 2-5). A Polaris 4x4 ATV is modified for this purpose. Vehicle's steering, throttle, brake and gear systems are modified yet its conventional driving system is conserved. It has waypoint navigation using obstacle avoidance and anti-roll sensing. Redcar Scout is equipped with a thermal imager, image intensifier and low-light CCD cameras.

Figure 2-5: Redcar Scout

In another similar conversion a Polaris Sportsman-500 ATV has been chosen as a base (Figure 2-6) [8] and converted into a UGV called IZCI that is to be used as a research platform. The conversion is done in such a fashion that, the vehicle is still manually usable, and a user can always overtake the vehicles control. In this conversion process, the systems (steering, acceleration control, gear shifting and brake) that are on the original vehicle have been modified. The original Ackerman steering system on ATV has been modified by adapting a DC motor to the chassis of the vehicle. Output of the DC motor has been transferred to the steering rod using a timing belt with two sided teeth.


Figure 2-6: İZCİ, views from both sides

An RC servo motor has been adapted to the gas throttle with a lever arm and a cam and can be used in parallel with the throttle control located on the handlebar. A solenoid has also been added to the system to limit the rotation of the lever arm for increased safety. The original gear shifting mechanism has five stages; park, rear, neutral, low and high. Modification of the transmission actuation has been achieved using a linear actuator which replaced the gear shifting rod. In the new breaking system, a cam driven by a DC motor actuates the hydraulic brakes while the existing braking system remains on the vehicle.

Team MonsterMoto has participated in DARPA Grand Challenge with their vehicle “JackBot” which is based on a 2004 Kawasaki KFX700 two wheel drive ATV

(Figure 2-7) [9]. It has liquid-cooled 697 cc, four-stroke engine with continuously variable automatic transmission. A large servo motor has been mounted on original steering column to achieve the steering action. Small servo motors have been attached to existing throttle and brake cables to perform these actions by wire. Additionally, a separate fail safe parking brake is driven directly by the emergency stop circuit.

Figure 2-7: JackBot

Team MonsterMoto implemented a single computer for processing tasks which is based on a 3.2 GHz processor with Windows XP operating system. For navigation purpose, vehicle had a Crossbow Navigation Attitude Heading Reference System (NAHRS) module and guidance application. A LIDAR system in front of the vehicle provides obstacle information about the environment.

This literature survey on development of UGVs shows that it is a common practice to use an ATV and modify it according to the needs of the projects. The main systems to be considered when such modifications are being made are steering mechanism, gas throttle, gear shifting and brake mechanism. Different methods and actuators are being used for these purposes. A summary of these conversion methods are listed in Table 2-1.

Table 2-1: A summary of methods used to convert an ATV to a research platform.

| **Vehicle** | **Throttle** | **Brake** | **Gear** | **Steering** |
|---|---|---|---|---|
| CyberScout [3] | Servo Motor | Hydraulic | Hydraulic | Hydraulic |
| Spirit [5] | Servo Motor | Servo Motor | Servo Motor | DC Motor |
| Ensco [4] | Servo Motor | Servo Motor | Relay | Servo Motor |
| Amor [6] | Mag. Valve | Mag. Valve | Mag. Valve | DC Motor |
| JackBot [9] | Servo Motor | Servo Motor | Linear Act. | Linear Act. |
| Overboat [10] | Servo Motor | Servo Motor | Servo Motor | Servo Motor |
| Cajunbot [11] | Servo Motor | Servo Motor | DC Motor | DC Motor |
| İzci [8] | Servo Motor | Servo Motor | Linear Act. | DC Motor |

Since all the ATVs introduced in this survey rely on internal combustion engines, servo motor is commonly selected to control the throttle valve. Similarly brake systems of the vehicle are controlled using servo motors due to its position sensing capability. Different applications show that there are different means of controlling gear shifting chosen by research groups. One of the groups used hydraulic systems for steering mechanism however most of the applications implement high power DC motors in this purpose.

Most practical way to begin conducting research on autonomous vehicles area is to modify steering, brake and throttle systems of an existing vehicle or use some parts of it and develop a new chassis based on those parts. By this way, one can come up with a drive-by-wire vehicle in a faster way than designing and developing a vehicle body from the scratch. The provided literature survey is in agreement with these statements. Hence, similar conversions are practiced by several research groups. Converting an existing vehicle into an autonomous one is justified based on couple of reasons: first, these vehicles have been tested and approved to operate on challenging environmental conditions, second, due to mass production their prices are very reasonable (especially in comparison to robots of comparable sizes) and finally, building a robust platform from scratch is a challenging research on its own, hence starting up with an existing frame significantly reduces the time required to build the platform on which developed algorithms will be deployed.

#### 2.2.1.2 Power

Power issue is important to the unmanned vehicle research since is the main factor that defines the range, operating time, speed and acceleration of the vehicle. The means that supply power to the system should not bring excessive weights and volumes to the platforms while maintaining reasonable operating time and related criteria.

Team ENSCO used the stock 649cc single cylinder four-stroke engine as the main power supply. The engine produces nearly 25 kW power. As a side power supply they maintained seven sealed lead acid batteries. The batteries were rated at 12V and 20Ah. The main purposes of these batteries were to power up the engine and lights. Sealed lead acid batteries are chosen to due to their rugged casing.

In the vehicle JackBot, besides the liquid-cooled 697cc four-stroke internal combustion engine, a 12V generator is used with an additional 24V, 65A alternator to power up the computing, sensing and actuating systems.

University of Florida had a program for developing a platform to study autonomous navigation technologies [12]. The power system of the vehicle named NTV2 relies on a 1000 Watts gas powered generator as the primary source of electricity. They used Honda EU1000i generator since it supplies stably conditioned power for use with sensitive electronics (Figure 2-8). A gas powered generator was selected over a battery based system because it can operate over extended periods of time with an adequate fuel supply, and there is no down-time for recharging. The AC power output from the generator supplies a 940 Watt Uninterruptible Power Supply (UPS) which provides battery back-up when the generator is off. While indoors, the UPS is plugged into an extension cord from a wall outlet.


Figure 2-8: Honda EU1000i Generator

The electrical system of Alice [13] has a 120 VAC, 3000 W generator which supplies two 1.5 kW chargers. Excess energy is stored in the battery system which includes 210 Ah, 12 V gel batteries. Using these batteries, sensor and processing systems are powered. ARSKA of Helsinki University of Technology [14] had a generator in addition to the vehicles own 12 V batteries. The capacity of the generator is 24 V and 1 kW, which is more than enough for the PC system, but is needed for testing of additional systems. As energy buffers two serially connected batteries of 12 V are used. The 24 V power from the batteries is converted to different voltages with several DC/DC converters. CyberScout of Carnegie Mellon University relied on 4-stroke internal combustion engine of their stock Polaris Sportsmen 500 ATV. As auxiliary power source, 2500 W, 120 VAC generator is selected and installed on the vehicle. Power system has voltage levels of 5 V, 12 V and 24 V.

Table 2-2: A summary of power sources used on ATV-based research platforms.

| **Vehicle** | **Main Power Source** | **Auxiliary Power Source** |
|---|---|---|
| CyberScout | 498cc 4-stroke Engine | 2.5kW Generator |
| Ensco | 649cc 4-stroke Engine | 12V Lead Acid Batteries |
| Overboat | 499cc 4-stroke Engine | 3kW Generator |
| JackBot | 697cc 4-stroke Engine | 12V Generator |
| ARSKA [14] | 1000W Generator | 12V Lead Acid Batteries |
| NTV2 [12] | 1000W Generator | N/A |

It is seen that preserving the stock vehicles' internal combustion engines is the common practice among these research. Since most of the stock vehicle used as a base includes gasoline engines, with its stock alternator, energy necessary for the electronic assets are maintained. Besides the engines, vehicles generally include deep cycle batteries or generators working with gasoline. These energy sources are auxiliary, since they are supplement to the main source.

#### 2.2.1.3 Processing

With development of sensor and interface technology data flow from sensor hardware to the controller units are increased to a level that powerful computers needed to handle this information. Also, with much more central processing unit (CPU) power compared to past, more resource consuming algorithms are developed and used in sub systems like mapping, localization or navigation. In this aspect, processing and storage power become an important criteria for performance of the robots. Powerful CPU units are installed on modern research platforms with larger random access memories (RAMs) in numbers of one, two, three or even more depending on the distributed processing technique used in the applications. Also to provide an interface for the peripherals, data acquisition boards are used commonly.
In CyberScout project, the computational architecture is divided into two categories: low-level and high-level processing. Low-level processing which includes locomotion task is performed by a PC/104 computer. In addition, high-level processing is performed by three PCs which are interconnected via network. PC/104 is selected due to its small size, low power requirement and rugged design for rough conditions which make it suitable for outdoor and off-road applications (Figure 2-9). PC/104 configuration employed includes four boards. Since the CyberScout project is a bit out dated (i.e. 1999) the processing powers of the computers are low compared to modern PCs. But the configuration is still important reference for other projects. One of the boards is Versalogic VSBC-2 CPU and the other one is WinSystems PCM-COM4A serial interface board. WinSystems PCM-COM4A is a module board with RS-232, RS-485 and RS-422 supports. Another one is WinSystems PCM-FPVGA video board. It has VGA CRT. The last PC/104 is DM5416 analog/digital I/O board with 16 analog and 8 digital inputs and outputs.

High-level processing tasks are done by 3 PCs with Pentium II 350MHz processors. The three PCs are interconnected with network.

Figure 2-9: PC/104 Rugged Casing

The computing platform of Alice, DARPA Grand Challenge 2005 participant, has six server computers each having 3 GHz CPU and another computer with 2.2 GHz CPU. All computers are linked together via Ethernet interface. As operating system, a Linux based system is chosen.

Another Grand Challenge participant, JackBot had a computing system has a computer with 3.2 GHz CPU. Computer includes Windows XP operating system. ARSKA of Helsinki University of Technology is equipped with a PC with 80486 processor is used for the control of the vehicle. The PC is equipped with two Analog & Digital I/O boards and a RS232 additional board. The RS232 ports are used for gyro, modem communication, ultrasound sensors and DGPS. The control of the throttle and steering is done with servo controller cards.

Team ENSCO's vehicle has three computers in the processing system. Two computers have a CPU of 1.6 GHz clock speed and the other one has 1.3 GHz CPU. They are implemented in a distributed schema which deals with environmental sensing, mapping and path planning, and vehicle control. PCI extension ports are

readily available for additional board. Data acquisition and filter boards are mounted on the computers via PCI slots.

Team Spirit of Las Vegas divided their computing systems into two: Vehicle Control Monitoring System Computer (VCMS) and Navigation and Video Processing Unit (NVPU). VXMS is a standard laptop computer with 3.2 GHz processor, 1024 MB RAM, 60 GB hard drive, 4-USB ports and 10/100T Ethernet connection. On the other side, NVPU consisted of Apple G5 2 GHz dual processor, 320 GB hard disk storage, 8 GB RAM, 1394 IEEE Firewire interface and 10/100T Ethernet connection.

Table 2-3: A summary of types and numbers of computers used.

| **Vehicle** | **Computer Type** | **Number of Computers** |
|---|---|---|
| CyberScout | PC/104 - PC | 4 |
| Ensco | PC | 3 |
| Spirit | PC | 2 |
| JackBot | PC | 1 |
| ARSKA | PC | 1 |
| İzci | PC/104 | 1 |
| Alice | PC | 7 |

It is seen that the number of computers used depends on the processing system design. Generally, distributing the processing load to computers more than one brings the advantage of shorter processing time and independent processor units for different subsystems. PC/104 is chosen over a regular PC due to its compact size and rugged case design especially for outdoor applications. But a PC with a rugged and well-designed casing provides similar or better performance as PC/104.

#### 2.2.1.4 Environmental Sensing

Environmental sensing is a crucial feature of an unmanned vehicle. Especially for applications that include full or semi-autonomy, sensing the environment is the key for a successful motion planning and guidance. For these purposes sets of sensors are

utilized on research platform and the useful information taken from these sources are supplied either to a human operator or to the robots' controller systems.

Most commonly used sensor types in ATV and car-like projects are Light Detection and Ranging (LIDAR) sensors and cameras. LIDAR is an optical remote sensing technology that can measure distance and direction to an object by emitting light, generally pulses from a laser. LIDAR systems used in research of unmanned vehicles involve a laser range finder reflected by a rotating mirror (Figure 2-10). These LIDAR systems are 2-D planar scanners. They can be used primarily for obstacle detection. However in many applications, these 2-D scanners are tilted in a certain frequency to supply enough resolution resulting in 3-D laser scanned map of the environment.

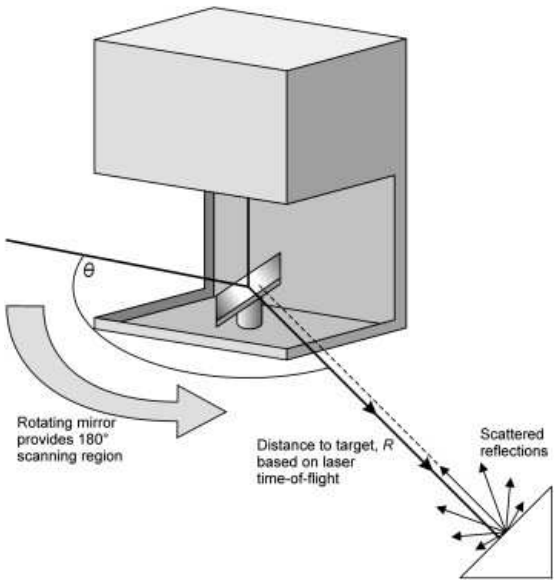


Figure 2-10: Working principle of 2-D LIDAR systems

Cameras used in research platforms are generally divided into two categories: mono and stereo vision cameras. Mono cameras are primarily used for surveillance or driving cameras for remote operators. Stereo vision cameras provide depth information. Therefore they are used to detect obstacles, map the environment or

provide the remote operator more depth enhanced vision for better teleoperation or telemanipulation.

Figure 2-11: Bumblebee2 stereovision camera by PointGrey

In Alice project, LADAR systems and stereoscopic camera systems are used in conjunction to avoid collisions (Figure 2-11, Figure 2-12). These sensor sets are used as complementary to each other especially in dynamic environment conditions. Also by this way if a sensor fails to gather enough information from the environment, other set compensates this failure. Sensors used in project are listed in Table 2-4.


Figure 2-12: Alice project and sensor locations

Sensors on the Alice are mounted in such a way that they provide necessary environment information for motion planning and obstacle avoidance. Two planer

laser scanners are placed on front bumper in different pitch angles. One is parallel to the ground while the other is placed in such a fashion that it points 3 meters in the ground from the front of the vehicle.

Table 2-4: Environmental sensors used in Alice [13] .

| **Sensor Type** | **Mounting Location** | **Specifications** |
|---|---|---|
| LADAR (SICK LMS 221-30206) | Roof | 180◦ FOV, 1◦ resolution, 75 Hz, 80 m max range, pointed 20 m away |
| LADAR (SICK LMS 291-S14) | Roof | 90◦ FOV, 0.5◦ resolution, 75 Hz, 80 m max range, pointed 35 m away |
| LADAR (Riegl LMS Q120i) | Roof | 80◦ FOV, 0.4◦ resolution, 50 Hz, 120 m max range, pointed 50 m away |
| LADAR (SICK LMS 291-S05) | Bumper | 180◦ FOV, 1◦ resolution, 80 m max range, pointed 3m away |
| LADAR (SICK LMS 221-30206) | Bumper | 180◦ FOV, 1◦ resolution, 80 m max range, pointed horizontally |
| Stereovision Pair (Point Grey Dragonfly) | Roof | 1 m baseline, 640x480 resolution, 2.8 mm focal length, 128 disparities |
| Stereovision Pair (Point Grey Dragonfly) | Roof | 1.5 m baseline, 640x480 resolution, 8 mm focal length, 128 disparities |
| Road-Finding Camera (Point Grey Dragonfly) | Roof | 640x480 resolution, 2.8mm focal length |

Three other LADAR systems are mounted on the roof of the vehicle to provide sensory information from 20, 35 and 50 meters ahead of the vehicle. In addition to the LADARs, stereovision camera pairs are used to provide depth information in 3-D fashion. Therefore obstacle detection and mapping systems become more robust and failure-safe.

Figure 2-13 shows Alice’s sensor coverage. In this figure, the small box represents the vehicle, Alice. Wide and narrow cones are the coverage of short and long rage

stereoscopic cameras respectively. Lines are the intersections of laser range finder readings with the ground.

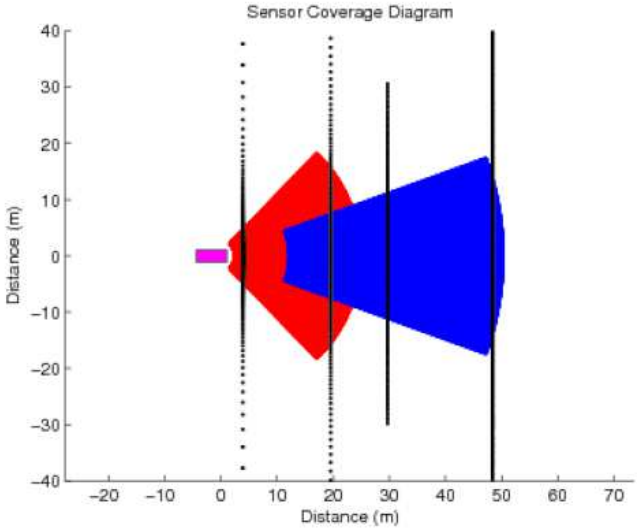


Figure 2-13: Coverage range of sensors mounted on Alice.

The environmental sensing system deployed in Team ENSCO's DARPA vehicle is composed of LIDAR, Doppler radar and stereoscopic camera. The vehicle avoids collisions with aids of LIDAR system and the stereovision camera. Velocities of the platform are measured by radar system located on the vehicle. Map of the environment is constructed with use of LIDAR and stereovision systems.

JackBot of Team MonsterMono from DARPA Grand Challenge 2005 used 4 SICK LMS 291 LIDAR systems mounted to the vehicle. All the sensors were mounted horizontally (i.e., 90 degrees to the ground). These sensors are just used to detect upcoming obstacles rather than mapping the environment.


Figure 2-14: SICK LMS 221 LIDAR

CajunBot from University of Louisiana was another competitor in DARPA Grand Challenge 2005 [11] (Figure 2-15). For the environmental sensing for autonomous navigation, they implemented LIDAR systems on the vehicle. Two types of laser range finders are used: SICK LMS 291 LIDAR and SICK LMS 221 (Figure 2-14). With these sensors, CajunBot is able to detect obstacles and avoid them.


Figure 2-15: CajunBot

Team Overbot uses many kind of sensor for environmental sensing. To avoid collisions with other vehicles, an Eaton VORAD radar system is mounted in front of the vehicle. It is connected to the computer via serial line. A SICK LMS 221 LIDAR system is installed with a pitch angle on top-front of the vehicle. It detects road profile with 2-D scan readings. Vehicle is able to follow a road by means of a

camera. The digital camera used was a Unibrain Fire-I 400. It has a resolution of 640x480 pixels. Additionally, ultrasonic sensors are used. Scanning area of ultrasonic sensors surrounds the vehicle. These ultrasonic sensors can sense an object at most at 3 meters from the vehicle so they are used especially for low-speed driving conditions.

It's seen in the literature that using planar LIDAR systems is a common practice for sensing the environment. In some of the projects, these LIDAR systems are tilted with help of an electric motor. By this way, 2-D slices of the upcoming environmental terrain and obstacles are gathered. After sensing of a complete angular range, these slices are combined to obtain 3-D information about the environment. Due to computation complexity of this method, in some other projects more than one laser scanners are placed in different fixed angles to provide rough estimate of the position of the obstacles and terrain shapes. An alternative solution to these approaches is using 3-D laser scanners. An example to these scanners is Velodyne HDL-64E 3-D LIDAR system (Figure 2-16). But its high prices, makes researcher to develop other techniques described above for 3-D environment sensing.

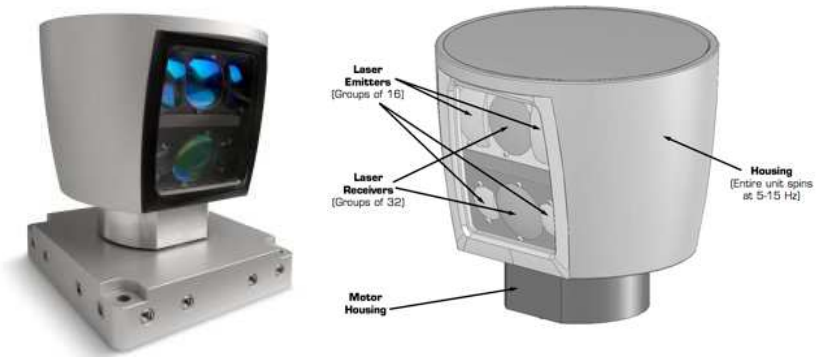


Figure 2-16: Velodyne HDL-64E 3-D LIDAR

To add supplementary environmental information to the laser scans, stereovision cameras are used widely. Depth information is taken from these cameras due to its

displacement between cameras mounted on. In addition to depth information, also texture sensing is done to supply necessary images to image processing engines.

Table 2-5: A summary of numbers of sensors used for environmental sensing

| **Vehicle** | **Number of LIDARs** | **Number of Cameras** |
|---|---|---|
| Alice | 5 | 3 |
| Ensco | 1 | 1 |
| CajunBot | 5 | None |
| JackBot | 4 | None |
| İzci | 2 | 2 |
| Overbot | 1 | 1 |
| CyberATV | None | 5 |

Since most commonly used environmental sensors are LIDARs and camera systems, summary table is formed based only these two. But it is seen that other sensing technologies such as ultrasonic range finders or RADAR systems are also used in similar projects.

#### 2.2.1.5 State Sensing

Team ENSCO used an inertial navigation system (INS) in the vehicle for state sensing purposes. The device had three accelerometers and three rate gyroscopes that give the measurements according to the vehicle inertial state. The sensor provides six degrees of freedom information, three for translations on three axes and three for rotation. Inertial state of the vehicle is measured with INS, but vehicle state (whether engine is running or not, temperature rise etc.) is also monitored by various sensor systems located around the vehicle.

In Alice project, an inertial measurement unit (IMU) is used as the primary state sensor of the vehicle. As IMU, Northrop Grumman LN-200 is used (Figure 2-17).

Figure 2-17: Northrop Grumman LN-200 IMU

The LN-200 comprises three gyroscopes and three accelerometers. Gyroscopes are produced with solid state fiber optic (FOG) technology. Accelerometers are made up using MEMS technology. These two systems are included in a rugged casing. The sensor measures acceleration values in three dimensions, and rotation speeds around these three axes. It outputs the information via digital output channel which occupies a serial communication link.

Team Spirit of Las Vegas used a variety of sensors to monitor vehicle state. A three axis gyroscope from Microstrain is used. With aid of this sensor, rotational speed values in three axes and the heading of the vehicle is determined. In addition to gyroscope, Honeywell altitude sensor is used with GPS data to precisely measure the altitude of the vehicle. Also, temperature monitoring of engine and other critical parts are provided with use of temperature sensors around the vehicle.

Figure 2-18: Oxford TS RT3000 inertial navigation system

In Overbot, actuators like steering and braking actuators have their own feedback systems. Besides, speed measurements from engine and driveshaft are obtained. Velocity of the vehicle is measured with use of a Doppler radar system. Inertial measurements and state estimations are done with the measurement taken from inertial navigation system and magnetic compass mounted on the vehicle.

CajunBot implements an inertial navigation system from Oxford TS. Its model is RT3000 (Figure 2-18). It is used to measure inertial state of the vehicle with considerably high precision. It is mainly developed for aerial applications. It also includes some mathematical algorithms for better state estimation purposes.

Table 2-6: A summary of sensors used for state sensing

| **Vehicle** | **Accelerometer** | **Gyroscope** | **Compass** |
|---|---|---|---|
| Alice | Yes | Yes | No |
| Ensco | Yes | Yes | Yes |
| CajunBot | Yes | Yes | No |
| Spirit | Yes | Yes | Yes |
| Overbot | Yes | Yes | Yes |
| CyberATV | No | No | No |

It is seen in the literature that use of accelerometers and gyroscopes is a common application for better state sensing purposes. These two units generally combined in an IMU or INS package as a solution. In some of the projects a separate compass is used for heading estimation while in some others (like CyberATV) the heading sensing relied on GPS data.

#### 2.2.1.6 Localization

Localization is another key function of unmanned vehicles. The vehicle should have to know or at least estimate the location of itself in the global aspect to plan its upcoming motions. For outdoor applications, global positioning system (GPS) is nearly the only solution for global localization. Some enhanced systems of GPS like

Differential-GPS (DGPS) are also used for better accuracy and precision of localization predictions. CyberATV uses a NovAtel DGPS unit for localization purposes. Due to use of DGPS instead of pure GPS allowed the vehicle to sense its position by a 20-cm resolution.


Figure 2-19: Navcom SF-2050 (left) and NovAtel DL-4plus (right)

In Alice project, two GPS units are used simultaneously. One of the GPS modules was Navcom SF-2050 with 0.5 meters Circular Error Probability (CEP) and 2 Hz update rate (Figure 2-19, left). The second GPS unit was NovAtel DL-4plus with 0.4 meter circular error probability (CEP) and 10 Hz update rate (Figure 2-19, right). JackBot of Team MonsterMoto has both GPS and inertial navigation systems. These systems are combined in a single module named Navigation Attitude Heading Reference System (NAHRS) from Crossbow (Figure 2-20). This unit implements an Extended Kalman Filter, and filters the GPS and inertial measurement with this filter.

Figure 2-20: Crossbow Navigation Attitude Heading Reference System

Team ENSCO used a GPS receiver from Novatel. It has both L1 and L2 frequency capability. GPS signals from this device are combined with the Differential GPS (DGPS) signals that can be perceived either from commercial Omnistar HP or public Wide Area Augmentation System (WAAS) depending on the selection on the device. With DGPS support, localization measurements become more precise that can allow the vehicle move with more confidence. This global location information gathered from GPS system is integrated into inertial measurements and Kalman filtering is applied for better location estimation. Team Spirit also uses a similar localization system. It combines a DGPS system which takes information from WAAS and combines it with inertial sensor readings.

It is seen in the literature that primary solution to localization problem in outdoor applications is use of GPS and DGPS systems. When GPS signals are not available, dead reckoning is used with environmental sensors to update position estimates.

#### 2.2.1.7 Communications

Teleoperated vehicles need a reliable line of communication between the vehicle and the remote operator. Also partly or fully autonomous robots need this communication to make the control center monitor vehicle movements and decisions on an assigned task. Lack of robustness in communication causes time delays and low update rates. These negative effects decrease operation performance of unmanned vehicles even causing fatal errors. Communication link of CyberATV between the platform and the control computer is established with Wireless LAN (WLAN) connection over 915 MHz Wavelan technology while Team Ensco designed the vehicle with no communication with a remote base. In Spirit for a telemetry system is designed for testing purposes. This telemetry system is composed of a serial link, a mobile phone and a remote control computer.

NTV2 of University of Florida provided wireless connectivity between the onboard computers and remote development systems, via commercially available 802.11g Ethernet broadcasting equipment. The IEEE 802.11g standard operates in 2.4 GHz range and transmits data at speeds up to 55 Mbps. A D-Link wireless Ethernet access

point (model DWL-2000AP) was installed in the electronics rack and connected to the Ethernet switch. This wireless link also allowed tele-op mobility control of the vehicle with an Operator Control Unit laptop equipped with a wireless Ethernet card and joystick.

### 2.2.2 Software

Besides the hardware implemented in unmanned vehicles, software is an important aspect for proper control and navigation. A modular and powerful software framework should be developed or readily available frameworks should be used. On top of this framework sensor abstraction layers and controller algorithms should be implemented.

In CyberATV, there are three stages that constitute the control architecture (Figure 2-21). CyberRAVE, which is the highest level of the architecture, performs missions given in phrases like “explore the environment”.

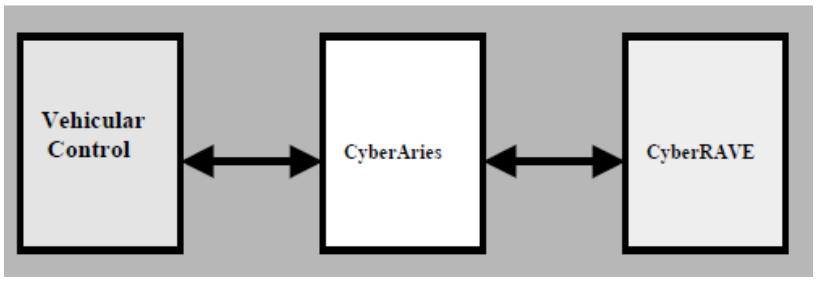


Figure 2-21: Control architecture design of CyberATV

Vehicle control system drives the steering and braking motors, it controls the position of the throttle motor to adjust the speed of the vehicle and it monitors the implemented navigation systems on the vehicle. Autonomous Reconnaissance and Intelligent Exploration System (CyberAries) is the link between vehicle control system and CyberRAVE.

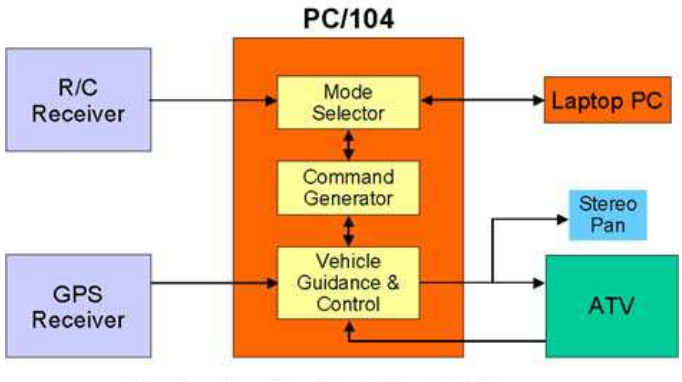


Figure 2-22: Vehicular control block diagram of CyberATV

The team behind CyberATV aimed to develop a control architecture that allows a modular controlling system. Vehicle control system is implemented on the PC/104 mounted on the vehicle while CyberAries runs on Pentium computer. Remote control computer has CyberRAVE installed in it and connected to vehicle via the communication link of 915 MHz frequency. In the vehicle, CyberAries and Vehicle control system are linked with serial communication line with RS-232 protocol.

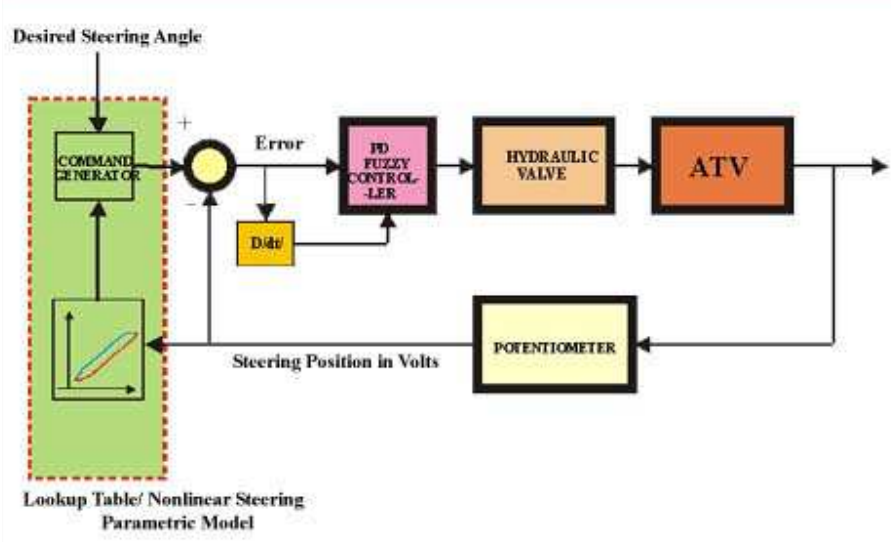


Figure 2-23: Block diagram for CyberATV steering controller

A fuzzy PD controller is implemented for steering control (Figure 2-23). It is selected because it is seen that it can produce considerable smooth steering commands that points the vehicle in accurate directions according to navigation system.

Speed controller is designed in such a manner that it allows moving smoothly at fairly low speeds (Figure 2-24). Low speed navigation is required because on-board camera systems need an amount of time for real time image processing.

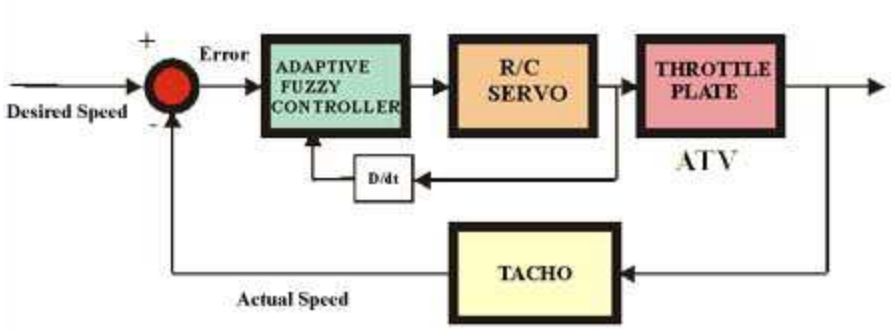


Figure 2-24: Block diagram of Speed control of CyberATV

CyberAries had four basic functional blocks: Perception, Mission Planner, Distribution Layer and World Model (Figure 2-25). Blocks (agents) are independent processes. Developer user can write a block according to the needs of the system and can run it as needed.

Each of the four agent blocks of CyberAries is made up of a collection of agents, thus offering a distributed and decentralized system in which agents run in parallel, concurrently and asynchronously, performing their own sensing (stimulus sources) and computation to command distinct outputs to other agents or actuators.

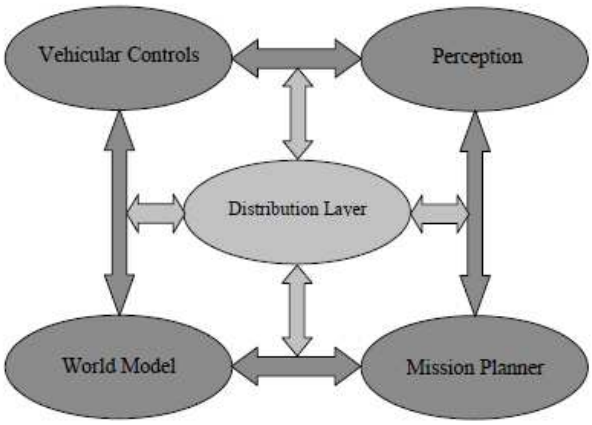


Figure 2-25: CyberAries Block Diagram

In another project, Alice, a general control system architecture is developed (Figure 2-26). LADAR and stereo vision systems that are used as environmental sensors are used to create a map of the environment where the robot operates. The information gathered from these sensors is combined with other state estimation data and the map in the global reference frame is created.

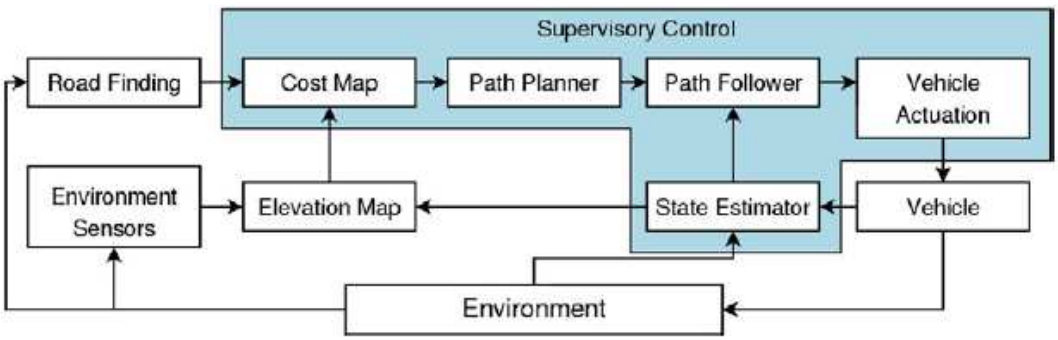


Figure 2-26: General system architecture of Alice

Created map has grid structure. Each grid has a value which represents the elevation of that location of the map. Map is centered on the vehicle. As the vehicle moves,

new grid cells are created. A cost map with speed limits is generated using this elevation map. By this approach, vehicle runs slowly in rough terrains and faster in easy-to-move terrains. This map is used by path planner process. Path planner estimates an optimal path considering the time required to travel. This generated path is sent to path follower process which can generate necessary mobility commands to the steering, braking and throttle actuators.

The CajunBot has an Autonomous Guidance System. It is written with C++ programming language and run on a Linux distribution, Fedora. There are some processes each responsible from another task on the vehicle. If the processes are on the same computer, they communicated via queues in memory. For other computers, subscribe-publish method is used. By this architecture, multiple computers can be implemented on the control system. Collision avoidance system has a path planner running. Path planner decides the path to travel and the steering process of the system generates necessary steering commands which consist of steer and speed values. For a left turn, positive steer value is used. For a right turn a negative one.

Besides these application and condition specific software solutions, there are some open-source projects that form a framework for robotic applications. The Player project (also known as Player/Stage) is one of these projects. It was founded in 2000 by researchers at University of Southern California at Los Angeles. The software of Player is distributed under GNU License. The Player has two sub-projects as the Player and the Stage. Player is a networked robotics server while the Stage is a 2-D robot simulation environment. The Player software can run on Microsoft Windows, Linux, Mac OSX and similar POSIX compatible operating systems. Player software is the hardware abstraction layer. Hardware on the robot can be interfaced through Player software. Player has a support for hardware like LEGO, iRobot, MobileRobots robots and sensors like Hokuyo and SICK laser range finders, Microstrain and Crossbow IMUs. It also allows users to write their own codes in C/C++, Python and Ruby programming languages with aid of client libraries present in the software. The Stage is the simulation environment built on top of Fast Light

Toolkit. Stage supports 2D simulation with multiple robots. It can communicate with Player to supply necessary simulated sensor and hardware information.

Microsoft Robotics Developer Studio (MRDS) is an environment similar to Player/Stage project. It includes both robot control and simulation capability. It is operated under Microsoft Windows operating systems and aimed academic hobbyist and commercial developers. It can be programmed through its graphical programming tool. It has a 3D simulation environment and gives access to hardware information on the simulated robot. It is programmed mainly with C# programming language.

It is seen in the literature that, special applications and robotic controller frameworks can be developed by research teams. The biggest advantage of this kind of software is being application specific and customizable according to the needs of project. Software for this purpose can be developed in variety of platforms. Windows OS can be used for .NET framework based projects. But the literature suggests the use of open-source UNIX based platforms for more flexible and more affordable solutions. Also MATLAB environment can be used either in normal mode or with Real Time Windows Target (RTWT) Toolbox. Also without Windows or any other operating system, MATLAB xPC Target can be used. In the literature it is seen that open-source or commercially available robotic controller framework projects are also widely used. The Player project, ARIA or Microsoft RDS can be given as examples of this software.

### 2.3 Managing Workload and Increasing Human Performance in HRI

Human-robot interaction (HRI) is a very important aspect of applications especially for operations that take place in extreme conditions like military, space exploration or search and rescue missions. Since the operator workload is the main problem to be handled for the sake of teleoperation, in last decades HRI topic gained increased interest. The interaction between human and a robotic agent should be understood for development of successful applications. Although the technology used in robotic assets developed throughout the years human role stays as the most critical part of

missions. For example in commercial aircraft accidents history, preventable human error is the main cause of at least 80% of airplane crashes [15]. Human error cannot be factored out of the system so one must develop applications regarding the limits of human behavior.

Teleoperation of robotic assets are involved in various contexts. For example search and rescue mission, robotic surgery operations, space exploration or military mission can be listed [16] - [17]. This teleoperation missions can be carried out using different input devices ranging from personal digital assistant (PDA) [18] and cellular phones [19] to joysticks, steering wheels and pedals [20], [21]. Generally speaking, control stations have different systems to display sensory information, plans and commands, state of the robot, progress about the mission and map of the operation environment. If the user interface device is small like PDA, generally touch-based interactions are employed [18], [22].

Unmanned systems range from small toy-like sizes to large vehicles that have tons of weight [23]. U.S. Army's Future Combat Systems (FCS) division is one of the groups that study robotics agents of different size that can operate semi-autonomously or teleoperated [24]. Teleoperation is a part of a robotic operation even it has autonomy built in. It will require human interference at least at one point of the mission. For example, teleoperation will be necessary when semiautonomous systems encounter particularly difficult terrain including natural or manmade obstacles [25]. In some cases that require critical movement the algorithm in the robotic asset even can return the control of the system totally to human operator [26]. This section of literature survey examines human performance issues related to teleoperation, especially focusing on remote perception and navigation, and also surveys potential user interface solutions to enhance teleoperator performance.

Human performance issues involved in teleoperating unmanned systems generally fall in two categories: remote perception and remote manipulation [27], [28]. In [29], it is mentioned as "with manual control, performance is limited by the operator's motor skills and his ability to maintain situational awareness". In the teleoperating

environments, human perception is often compromised because the natural perceptual processing is decoupled from the physical environment. In teleoperation decoupling of the operator from the physical environment affects human's perception of affordances in the remote scene and often creates problems in remote perception such as scale ambiguity [30]. Even basic tasks can be hard to perform for the teleoperator due to mismatch of visual and motion feedbacks in human body. Placing the camera at a height that does not match the normal eye height can cause such a problem [31]. Teleoperation in a polluted environment requires sensing of actual sizes of objects to decide the next maneuver during operation [32]. During teleoperation, human's object identification capability is degraded [33]. Inadequate video streams can cause poor situational awareness [34]. According to [35], robotic assets can malfunction or break down every 6 to 20 hours during operations. This causes low reliability. The following section discusses in detail how remote perception and manipulation is affected by factors such as limited field of view (FOV), orientation, camera viewpoint, depth perception, degraded video image, time delay, and motion.

### 2.3.1 Multiple Resource Theory

Teleoperation missions may include tasks as navigating a robotic agent to a point, processing sensory information, communicating with others involved in operation or manipulating a remote object (telemanipulation). During these tasks, humans are subject to multiple incoming resource channels. Wickens and colleagues described Multiple Resource Theory (MRT) which will be used to describe the human-robot interaction [36]- [37]. MRT proposes that the human operator does not have one single information processing channel (Figure 2-27). Instead it suggests that several different pools of resources that can be tapped simultaneously.

The first dimension of the model is named as *stages*. It is subdivided into following categories: perception, cognition, and responding. Wickens [36] says that the perception and cognition stages can be separated from response stage in terms of resources they use. As an example, Wickens suggests that verbally confirming a

command (i.e., responding) does not interfere with visual observation of the environment.

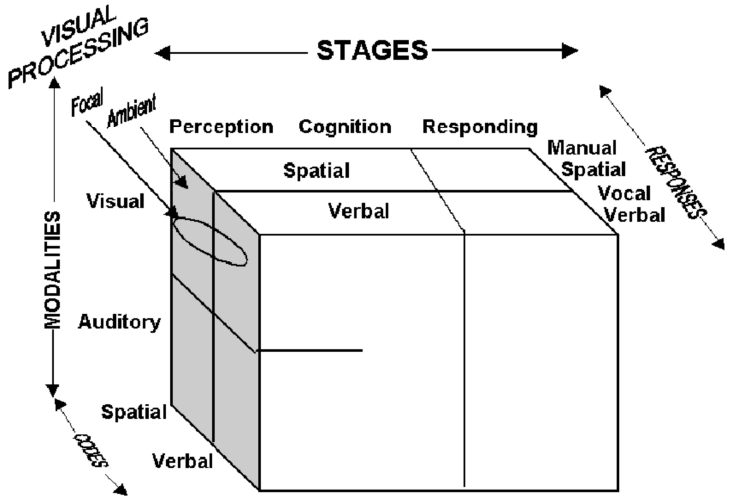


Figure 2-27: 3-D representation of the structure of multiple resources [37]

Another MRT dimension is called as *perceptual modalities*. It is stated that if the same modality is used for different sensory information, interference is more likely to occur. Therefore different sensor modalities should be implemented for different information channels. Using two auditory or two visual channels at the same time is defined as *intra-modal time sharing*. Besides, using one channel from each auditory and visual channel is named as *cross-modal time-sharing*. Wickens says that, dividing attention between the eye and ear (AV) is better than two auditory channels (AA) or two visual channels (VV) [36].

The final dimension, *processing codes*, defines the distinction between analogue, spatial processes and categorical, symbolic processes. There can be interference since both response and perceptual stages have processing. As an example, an operator may not communicate with team members while responding to alert especially in text format due to same symbolic processing requirements. The

operator may not even notice this interference. This section of literature survey describes the related studies on HRI workload within MRT model [36].

### 2.3.2 Effects of Visual Demands

Teleoperation, because of its nature, mostly relies on visual channels of the operator since this channel is used to navigate the vehicle in an environment and to inspect critical objects around. In the literature it can be seen that the research projects are trying to provide more realistic visual information with manipulating field of view and camera location. Also in these studies it is seen that by improving time delays, frame rates and depth information, operator's perception channels are employed better [27]. Therefore this part is divided into three topics as time delay, type of visual system and environmental complexity. In the time delay section frame rate and latency is inspected while camera type section involves field of view and camera location studies. Last section, environmental complexity is dedicated to depth information and environments crowded with irrelevant objects.

#### 2.3.2.1 Time Delay

Time delay is defined as lags in computing or transmission systems that affect display of information on operator interface. In some cases, time delay is unavoidable because the type of task does not allow it to be at a minimum rate. For example, since space exploration agents are apart from the operator in large distances, there exists an inherent lag which is unavoidable. Although it is better to reduce lags in the system for operator effectiveness, in cases as described above where inherent lags are present, limits and responsiveness of human behavior to the system delay should be investigated. In the literature it is common to manipulate frame rate (FR) or latency as system delay. Latency is the duration between the instant of an event to occur actually and its projection on a display system [27]. On the other side, frame rate is the number of frames from the display system in a time duration. It is generally specified as frames per second, fps.

In [38], seven different update rates starting from 0.5 Hz to 24 Hz are applied in a targeting mission for UAV. Results suggest that as the update rate increases performance ratings also increased. Massimino and Sheridan conducted a similar experiment with three different frame rates as 3, 5 and 30 fps [39]. They concluded that efficiency is improved significantly as the frame rate increased. There are other studies supporting these findings also [40], [41], [42].

Studies related to latency generally measured efficiency and task errors during teleoperation. Lane et al. stated that in their study about UGV simulation, increasing time delays causes the efficiency to decrease [43]. In another study [40], it is seen that system latency causes significant learning effects on the operator.

From the studies in the literature, it is seen that frame rates should be as high as possible for increased efficiency. About latency, it is seen that, latency should be eliminated from the system. If it is not possible, like space missions, it should be kept constant because the operators can adapt themselves to constant latency situations [44]. A summary of results are listed in Table A-1 and Table A-2.

#### 2.3.2.2 Type of Vision System

Vision systems are evaluated in terms of range, perspective and/or field of view. These factors are important due to their dominant effect on visual perception of the environment by the operator. Field of view is manipulated in many studies to measure its effects to efficiency, workload and task errors. In [45], narrow and wide FOV are compared. Results suggest that task completion is faster and efficiency is higher with wide FOV. Similar results are obtained by Pazuchanics [46] stating that widening the field of view improves the performance compared to narrow one. On the other side, Scribner and Gombash identified that widening the field of view may induce motion sickness on the operators [47].

Camera perspective is another factor affecting workload and performance. Lewis et al. suggested that using gravity references display systems provides better situational awareness and increases efficiency in comparison to vehicle fixed display [48]. Also,

[49], identifies the same results. Olmos et al. studies split screen and exocentric displays. They measured increased performance with consistent split screen applications than exocentric displays [50]. In another study [51], authors concluded that third person view decreases response time and results in fewer errors when compared to first person view. Overall results of similar studies are listed in Table A-3 and Table A-4.

#### 2.3.2.3 Depth Information and Environmental Complexity

This section is divided into two as depth information and environmental complexity. Studies on depth information address mostly vision systems with depth perception while environmental complexity studies deal with complexity of vision and number of objects in the area of concern.

Studies about depth cues are mostly focused on comparison of monoscopic (MS) and stereoscopic (SS) display systems. MS displays provide two dimensional image of the environment to the eyes of the operator without any depth information. Size of objects, shadows and similar cues may guide the operator to figure out distances [52]. On the other side, SS displays generally take visual information from two cameras located in a retinal disparity from each other to provide two different images to two eyes of the operator. Human visual channel processes these two images and extracts depth information as it does with natural visioning. In environmental visioning such as observing an environment MS displays are more appropriate while for inspection purposes like focusing in a smaller area SS displays assists vision perceptual resources according to Wickens [36]. Draper et al. [52] studied a comparison between MS and SS display systems and stated that SS displays performs better than MS ones but especially in difficult situations. Lion [53] also supports these findings in his study. There are also other studies with similar findings [54], [55], [47]. But Richards et al. [42] found that when other modalities like tactile of haptic modalities are present in the system there will be not much difference between these two visual display systems.

If there are irrelevant objects to the operator during a teleoperation mission, environmental complexity increases. As Wickens suggests [36], environmental complexity mostly affects focal vision of human operator. This results in decrease of target detection capability. Chen and Joyner [56] compared dense and sparse targeting environment and stated that if a high number of irrelevant objects are present around a target, targeting errors increased. Similar results are obtained by Hardin and Goodrich [57], Witmer and Kline [58], and Yeh and Wickens [59]. Another aspect of environmental complexity, display image color, is studied in [60]. It is concluded that color image display provided better target detection capabilities and increased efficiency compared to the grayscale image. As a result of literature on this topic, increased environmental complexity affects target detection durations negatively but it does not have an effect on hit rate of targets. Solutions to this problem may be reducing visual information or transferring information to other sensory channels. Studies on depth information and environmental complexity are listed in Table A-5 and Table A-6.

### 2.3.3 Improvement by Display Design

MRT concludes that types of resource channels are the main issue that affects the operator performance. Along these resource channels, visual sensory channel is the most important one and it limits the user performance mainly. HRI literature suggests that users have better understanding and manipulation in visually simple environments [56], [61]. Manipulating visual cues to decrease workload is effective as seen in the studies [54], [59]. Besides visual cues, auditory and tactile feedback should be incorporated to increase operator performance [62]. In following subsections, possible improvements to display design by reducing visual demands are described.

#### 2.3.3.1 System Latency and Frame Rate

Increasing frame rate and decreasing latency is the most straightforward improvement to the display design. When these improvements are performed, user will perceive more realistic images and less difference between native visual

perception and virtual one. The main guideline should be increasing frame rate to a reasonable level for human processing of information and eliminating delays in the system. But if it is impossible like space exploration missions, latency should be kept constant. Throughout the mission, the operator will gain experience and compensate the latency present in the system.

#### 2.3.3.2 Camera Perspective and FOV

Pazuchanics states that, contextual resources should be integrated with other features of interfaces to decrease workload on the operator [46]. Workload [63] and motion sickness [47] are the main factors that define the field of view to be used. Also, operators preferred different FOVs according to the type of task they are required to perform [64]. The best practice may be allowing the practitioners to allow choosing optimal field of view according to their measurement on task and its criteria. In the literature it is seen that third-person view is superior to first person and a gravity-referenced camera orientation is better than vehicle fixed one. These choices improve operator performance [51]. As a summary vision systems should employ a level of FOV from moderate to wide but should not cause motion sickness. Also a third person perspective and a gravity-based camera mounting should be selected.

#### 2.3.3.3 Depth Information and Environmental Complexity

In related studies it is evident that SS displays have advantages over MS displays. However, this difference can be eliminated is other modalities, such as auditory or tactile, are present. Therefore it can be said that SS displays should be implemented if possible. If not MS displays should be enhanced with other sensory modalities. Besides, frame rates of MS displays should be as high as possible. Elimination of environmental complexity and irrelevant objects from the perceptional range of the operator speeds up the operation especially in target detection.

#### 2.3.3.4 Use of Multimodal Displays

Displaying all necessary information on a single display due to limitations in hardware systems or nature of tasks involved can overload the visual sensory channel

of the operator resulting in excessive workload negatively affecting user performance. In such cases, other modalities should be used and implemented in display systems to relieve the visual channel of operator. By using multimodal displays, task information is provided to the operator in different sensory modalities like auditory or tactile. This improves operator workload which is a primary concern in HRI field. There are already a number of reviews in the literature that list the benefits of multimodal displays [65], [66], [67], [68]. According to these studies it is apparent that adding other modalities increases operator performance during teleoperation missions. Burke et al. [65] stated that tactile feedback is more effective than auditory feedback when mission criteria are high to accomplish. As a result multimodal feedback is found useful especially when poor visual conditions are present in the system [39], [42]. Auditory modality is seen effective to decrease reaction time since auditory sensory channel is more sensitive to system alerts [69], [62], [70].

#### 2.3.3.5 Possible Future Research

Improvements gathered by use of haptic and force feedback in human-robot interfaces especially for teleoperation missions are examined in relatively few studies surprisingly. Force feedback is used already in many tasks such as flight, entertainment, video gaming and simulations due to its benefits. Haptic interfaces can be used in teleoperation of a robotic agent but there are not many studies realized such an improvement for teleoperation purposes. Use of force feedback can decrease respond demands of tasks in robot operators significantly.

### 2.3.4 Effects of Response Demands

Especially human-critical (e.g. search and rescue) and military missions require rapid responses during missions. The user should act continuously and give quick responses to inputs for the sake of mission success. However human output channels are limited so the performance can be degraded when there are multiple tasks and high response demands. In the literature it is common to manipulate response demands to figure out limitations of humans. There are two manipulations mentioned

frequently in the literature: task standards and number of robotic agents to be controlled.

#### 2.3.4.1 Task Difficulty

Task difficulty is another factor that has major effects on response demands of teleoperators. Task difficulty may be defined by number of targets to hit, number of system alerts that user is required to respond, time pressure (i.e., limited time allowed to complete a mission), distance to be navigated and number of tasks assigned. In [71], authors changed the number of targets to hit and they concluded that low number of targets reduced the required workload on the operator. Cosenzo et al. measured the errors in targeting and response time to navigational decisions and they observed that increasing the number of targets also increased the targeting errors and reaction time [72]. Draper et. al stated that if the system alerts become more frequent, operator is distracted and his operation performance degraded [73]. Alerts should be provided the user but in a reasonable way that will not stress him above a certain level. In [25], effects of navigation distance to workload of the operator are investigated. Results of the study states that if the distance to travel with robotic agent is increased, workload also increases while line of sight is not affected the result significantly.

Both Mosier and Hendy observed the effects of time pressure on the response demands of teleoperators [74], [75]. Mosier et al. applied low and high time pressure and measured errors and efficiency in a flight simulator. They concluded that increased time pressure positively affected the pilot efficiency but worsened the rate of diagnosis errors. Similarly Hendy et al. observed that performance is negatively affected if only high time pressure is applied. But on the other side, very low time pressure induces workload on the operator.

According to these results, not surprisingly, increasing the task difficulty negatively affects the operator performance. Increased task difficulty results in high response demand which induces high workload on the operator. As a result, operator performance degrades. A summary is presented in Table A-7.

#### 2.3.4.2 Multiple Platform Control

Number of robotic assets to control has direct impact on the response demands of teleoperation tasks. One can estimate easily that, increasing the number of platforms to be controlled increases the operator workload and affects the teleoperation performance negatively. However, since multiple agents can reduce the task standards by sharing the total workload between them, they should be deployed in some scenarios. The trade-off between increased operator workload and advantages of having multiple agents should be identified.

Adams [76] changed the number of platforms to be controlled as 1, 2 and 4 UGVs. In the study, efficiency and workload for a search and transfer mission is observed. As a result, the study concluded that having 1 or 2 UGVs in control does not affect the measurements significantly but 4 UGVs considerably decreases efficiency and perceived workload. Similar to this study, Chadwick [77] observed that having 1 or 2 UGVs in control does not affect errors and perceived workload in targeting significantly. In [69], 1 and 2 unmanned aerial vehicles are used to measure errors in tracking and targeting. Results indicate that 1 UAV task induces less workload on operator than 2 UAV case. Same result is verified by Hill and Bodt [78].

Having more robotic assets to be controlled allowed the users to navigate more in distance overall and take more actions than in case of controlling one platform [79], [80], [81]. But at the same time, in [69] and [71], targeting and navigation errors are increased with the number of platforms. Besides, reaction times of the operators are increased which affected the task negatively [82], [83]. As a result, increasing the number of platforms allows users to take more action but at the cost of error rates and response times. Overall results are listed in Table A-8.

### 2.3.5 Improvement by Automation

Since human has limited response channels increased response demands results in degraded teleoperation performance of users. The solution to this problem should be reducing response demands of operations from the human operators. This can be

achieved by introducing a level of autonomy to the robotic agent controlled. At first glance, it can be assumed that increasing level of autonomy decreases the human workload and positively affects the operation performance. However, human-robot interaction includes more complex relations. Studies in the literature divided this topic into two as level of autonomy (LOA) and automation reliability. Each topic is investigated in its corresponding topic below.

#### 2.3.5.1 Level of Autonomy

Level of autonomy defines the sharing of operation and control tasks between human and robotic asset. There are studies in the literature which define the levels of autonomy but the most common used description is defined by Tom Sheridan [84]. Sheridan defined a scale from 1 to 10 as increasing LOA. Scale is given as following scale.

1. Computer offers no assistance; human does it all.
2. Computer offers a complete set of action alternatives.
3. Computer narrows the selection down to a few choices.
4. Computer suggests a single action.
5. Computer executes that action if human approves.
6. Computer allows the human limited time to veto before automatic execution.
7. Computer executes automatically then necessarily informs the human.
8. Computer informs human after automatic execution only if human asks.
9. Computer informs human after automatic execution only if it decides too.
10. Computer decides everything and acts autonomously, ignoring the human.

Bruemmer et al. studied the manual versus shared robot control while searching for targets [85]. They measured the efficiency and targeting errors. Results indicate that use of semi-autonomy by means of shared control improves the efficiency and reduces targeting errors. In [86], ten levels of autonomy are applied and it is seen that when human defines the options and automated system implements these options results are superior to other levels of autonomy. Hardin and Goodrich conducted a

search and rescue mission with varying levels of autonomy [57]. The study measured efficiency and workload of the operator and concluded that mixed autonomy between user and the platform performs better than a human or robot being in full control. In another study, Hughes and Lewis compared user controlled and sensor driven target-search UGV camera [87]. Results suggest that sensor-driven camera performs better in target detection than manually controlled one. Kaber et al. also measured situational awareness (SA) with manipulated level of autonomy [88]. Level of autonomy is changed from simple support to full autonomy and errors, workload and situational awareness are measured. The study concluded that increased LOA results in improved performance and reduced workload. However, it may cause loss of SA for some functions.

In [89], 3 levels of autonomy is implemented as manual control, veto-only control and autonomous waypoint navigation. Throughout the study efficiency, usability and error rates are measured. It is seen that increasing the autonomy improves efficiency and usability. Also it reduces the effects of system delays. In another study, Wickens et al. implemented UAV missions with full manual control, auditory aid and automated flight. Results suggest that automation improves target detection performance.

Results in the literature show that level of autonomy should be increased as high as possible. However it is seen that full replacement of human role with autonomy is not possible. A summary is presented in Table A-9.

#### 2.3.5.2 Automation Reliability

Another aspect of improvement by automation is automation reliability. Automation aids can be assumed to have superior effects on teleoperation performance of humans. However, studies in the literature show that it is the reliability of the automation aids which directly affects the workload of the operator.

Chen [90] provided users targeting aids with imperfect reliability. While higher level of automation increased the performance, false alert signals negatively affected the

workload of the operator. In [91], automated alerts with 100% reliability, 67% reliability with false alarms and 67% with misses are given to the operator. Response time, situational awareness and error rate in targeting and monitoring system are measured. The study concluded that false-alarms decreased the use of aids by the operators. Kaber et al. compared normal operation and unexpected automation failure case [88]. It is seen that in low levels of autonomy with more human control resulted in increased situational awareness and performance. Muthard and Wickens conducted flight simulation tests with or without reliable automation [92]. When flight plan is selected automatically, pilots ignored the environmental changes that made the flight unsafe. 60% and 80% decision reliability is implemented in [93]. It is seen that imperfect decision-making automation decreases the performance which is a result of operator complacency with automation system.

As a result from the literature, it can be concluded that reducing of workload can be accomplished by automating certain tasks. However, this is possible only the automation has nearly perfect reliability. If it does not have a high level of reliability, workload may increase in opposition to the intended result. A summary of results can be found in Table A-10.

# CHAPTER 3

# DEVELOPMENT OF AN UNMANNED GROUND VEHICLE

## 3.1 Introduction

Unmanned ground vehicles (UGVs) are becoming more and more important especially for human-critical missions in variety of application areas. These areas can be extended from military to search and rescue missions. In many research companies and universities around the world, ongoing research projects on this subject are present. Real experiment platforms, beyond the simulated environments, are indispensable for the sake of a research. Developing an unmanned ground vehicle (UGV) platform for semi and fully autonomous research projects is the main purpose of this section. Outdoor and mainly off-road working environment is chosen for the proposed UGV platform. Therefore ATV-like UGV development is carried out throughout the study.

## 3.2 Body Design

In the literature, it is seen that off the shelf All-Terrain Vehicles (ATVs) are chosen commonly as a base to the research projects. This selection is made based on the mobility capabilities of ATVs on unstructured roads and off-road environments. Throughout the literature survey it is noted that gasoline or diesel powered ATVs are converted into autonomous research platforms by modifying steering, braking, throttle and gear shifting mechanisms. Additional modifications are computer, power, communication and sensor systems deployed on these vehicles.

The selected ATV is electric powered. It has two in-wheel electric motors each rated at 3000W in the rear wheels. Unlike the similar projects in the literature, electric

powered ATV is chosen due to its silent operation in comparison with ATVs equipped with internal combustion engines. Its better low speed control and elimination of a need for separate power systems for engine and electrical systems are other reasons for this choice. For the development process of and UGV mentioned in this thesis, SynECO eRover 6000 - electric ATV is chosen as the base (Figure 3-1). Features of the stock vehicle are listed as below:

- Size (L x H x W): 203 x 107 x 112 cm
- Carrying capacity: 200 kg
- Weight: 200 kg (including 30 kg battery pack)
- Range per charge: 56 km
- Max speed: 50 km/h
- Climbing ability: 30 %
- Rated continuous output power: 6000 W
- Battery pack: 60V 40Ah LiFePO4
- Traction: 2 electric hub motors (rear wheels)

Figure 3-1: SynECO eRover 6000 Electric ATV

In addition to the choice of electric vehicle, development method of the UGV from selected ATV is the main difference of this project from the similar ones in the literature. Dislike the projects in the literature, the body of stock ATV is not used as it is. Instead, it is disassembled into its main subsystems and assembled back into a

custom designed chassis. Rear axle with two hub motors and wheels, front wheels, steering system, suspension systems and braking systems are taken from the stock ATV. A new main chassis is designed to provide necessary spaces for the hardware that are planned to be installed on the vehicle. The new chassis and body design of the UGV is shown in Figure 3-2.

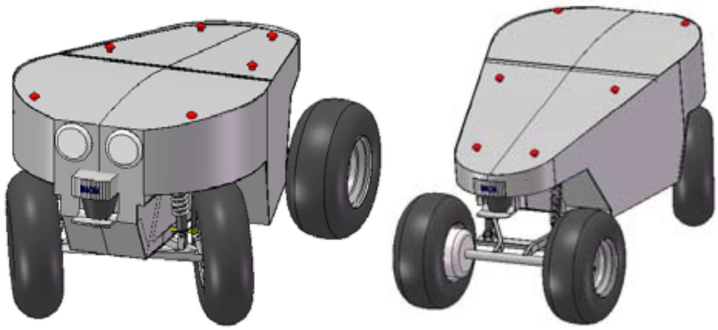

Figure 3-2: Redesigned main chassis and body of the ATV based UGV.

This redesign process mainly involves reassembling removed ATV subsystems (i.e. front wheels, steering, braking system, and rear wheels) and properly re-locating them within the new designed chassis. Front wheels of the vehicle are steerable and the rear wheels have electric hub motors installed in them. Therefore, the vehicle designed can be directed by using the differential drive of the rear wheels as well as steering of the front wheels. The main chassis is made up of steel while the body material is aluminum. The manufactured UGV is shown in Figure 3-3.

Figure 3-3: Developed and manufactured ATV

For night driving or dark area missions, two headlights are mounted on the front side of the vehicle. Each headlight contains 7 power LEDs that are driven with 24V voltage level. Safety and reliability is the main concern when a platform in such a size and power is designed. In an event of mechanical, electrical or software failure the UGV must come to stop as soon as possible. For this purpose, four emergency buttons are placed in various places on top of the vehicle. The buttons are placed in such a manner that anyone at any side of the vehicle can easily reach and press an emergency button to prevent further damage to the environment, human-beings or the vehicle itself.

### 3.3 Steering System

The steering system of the stock ATV is based on Ackermann steering principle that steers the front wheels in different angles to minimize the slip motion during cornering. This steering system from the original vehicle is conserved. The handlebar is removed; instead a DC motor is mounted and coupled with the steering hub via an R+W EK2 series high torque coupling A heavy duty permanent magnet geared DC motor having the specifications of 24 V DC nominal voltages, 1:250 gear ratio, and reversible rotation capability is chosen as the steering actuator (Figure 3-4, left). The motor is rated as 300 W. The motor has about 300 kg.cm continuous torque at the gearbox output while it rotates at 10 RPM which is sufficient to steer the wheels.

Figure 3-4: Steering motor from top (left) and rods from bottom (right)

The steering motor mounted in front of the vehicle is connected to the pin that is located on the intermediate part that joins steering rods (Figure 3-4, right). Rotational movement of electric motor is converted to pre-defined Ackermann angles in this section. The wheels have a steering angle +/- 30 degrees maximum. The turning radius of UGV is calculated regarding this value.

The definition of turning radius as given by the Society of Automotive Engineers (SAE) is as follows: “The distance from turning center to the center of tire contact with the road of the wheel describing the largest circle, while the vehicle is executing its sharpest practicable turn (usually to the outside front wheel)” [94]. An Ackermann steered vehicle is illustrated as turning in Figure 3-5. Turning radius depends on the wheelbase “a”, the distance “b” between the steering pivot axes, the maximum angle “ϕ” through the which the inside front wheel can be turned from the straight ahead position, and the scrub radius, “e”. Since,

$$R_T = d + e \tag{3.1}$$

$$d = \left[ b^2 + \left( \frac{a}{sin\emptyset} \right)^2 + \frac{2ab}{tan\emptyset} \right]^{1/2} \tag{3.2}$$

$$R_T = \left[ b^2 + \left( \frac{a}{sin\emptyset} \right)^2 + \frac{2ab}{tan\emptyset} \right]^{1/2} + e \tag{3.3}$$

For the designed UGV, these values are as follows:

$a = 1200\ mm$

$b = 880\ mm$

$\emptyset = 30°$

$e = 100\ mm$

With these values, turning radius for the vehicle appears to be,

$$R_T = 3.30\ m$$

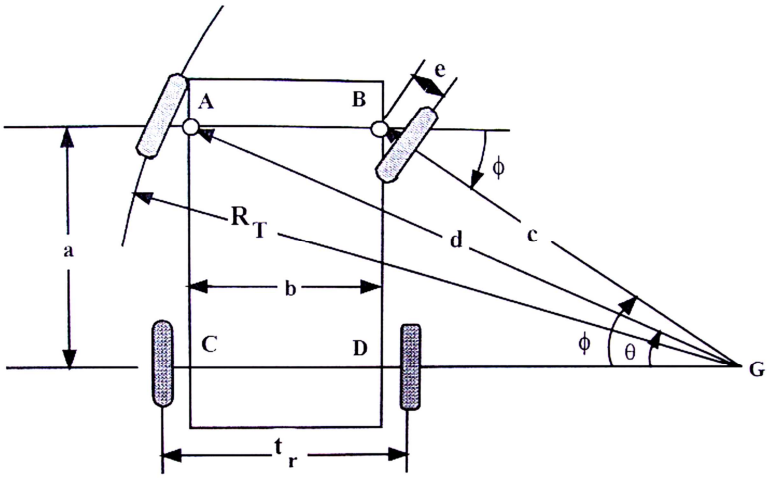


Figure 3-5: Turning radius illustration of an Ackermann steered vehicle

As a feedback device an encoder is mounted to motor shaft. SIEMENS 6FX2001-3CC50 2500 PPR quadrature incremental encoder is used. Since the start up position of the front wheels is unknown to the system, firstly a resetting routine is performed. This routine first turns the wheel in right most position up to the limiting sensor. At the limit position, encoder value is recorded. Same procedure is performed for the other direction. So the position control of the steering motor is performed. Steering system has two inductive proximity sensors. These sensors are activated when a turning limit is reached in whether clockwise or counter-clockwise direction. So the steering action is halted to prevent the steering mechanism from any damage (Figure 3-6).

For the control of the steering motor, digital servo controller, DPRALTE-020B080 by Advanced Motion Controls is used (Figure 3-7). DPRALTE-020B080 can output maximum power of 760W, has a supply voltage range from 20V to 80V while can supply current up to 20A. It has Current, Position and Velocity modes of operation and it can be commanded via +/- 10V analog voltage, 5V step and direction or serial commands using RS232/RS485 interface.

Figure 3-6: Steering and suspension mechanism at front wheels

In this application, DPRALTE-020B080 is commanded via analog voltage interface and the control loop is closed on the main onboard computer rather than the servo driver itself. This servo driver has also analog and digital input/output pins. Limiting proximity sensors from the steering mechanism are connected to the digital pins of the driver so in a limiting position the driver halts the operation and disregards the potentially dangerous insistent commands if there are any.


Figure 3-7: Advanced Motion Controls DPRALTE-020B080 motor controller

### 3.3.1 Steering Controller

Block diagram of the steering system is given in Figure 3-8. In this steering system, the desired steering angle is commanded to the control computer via an interface device. The steering wheel with force feedback capability is used as the interface in this study. Position of the steering motor shaft is measured with the industrial grade quadrature encoder whose pulse per revolution is 2500 which gives considerably high resolution. The encoder value is read via the data acquisition (DAQ) board mounted on the computer system.

This signal is compared with the desired value and the error is supplied to a generic PD controller. The command value obtained from the PD controller is converted back to analog voltage again with DAQ card.

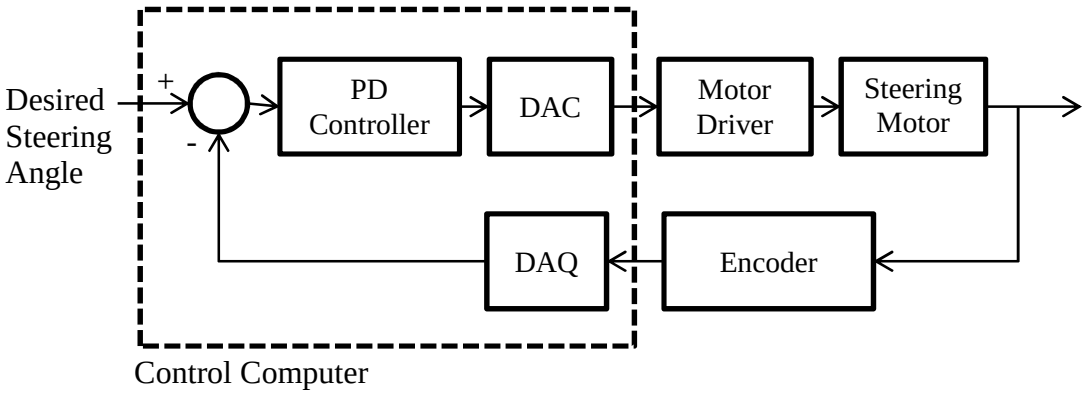


Figure 3-8: Block diagram of the steering controller of the UGV

This voltage is fed to the DPRALTE-020B080 motor driver and the driver adjusts the output voltage value that will be sent to the steering motor to position it with the desired steering value. From many available choices, a PD controller is designed to control the steering system. Since the steering system is the main factor that decides which way the vehicle moves a stable and reliable controller is a must for this application. There is no overshoot wanted in the response of steering angle which

would cause the vehicle to deviate from its desired path. A possible overshoot and further oscillations would even result in an unstable vehicle which will probably roll-over or damage itself or its environment. Because of these reasons, a simple proportional controller is not sufficient. Therefore, a PD controller is designed. Since the feedback device is a rotary encoder, feedback signal is free of noise and jittery which would cause the derivative term to not function properly. As a result, derivative term is applied and well functioned. Integral term is not needed since there is no significant steady state error in the step response. Also, integral term is not included since no overshoot is the primary constraint of the desired controller.

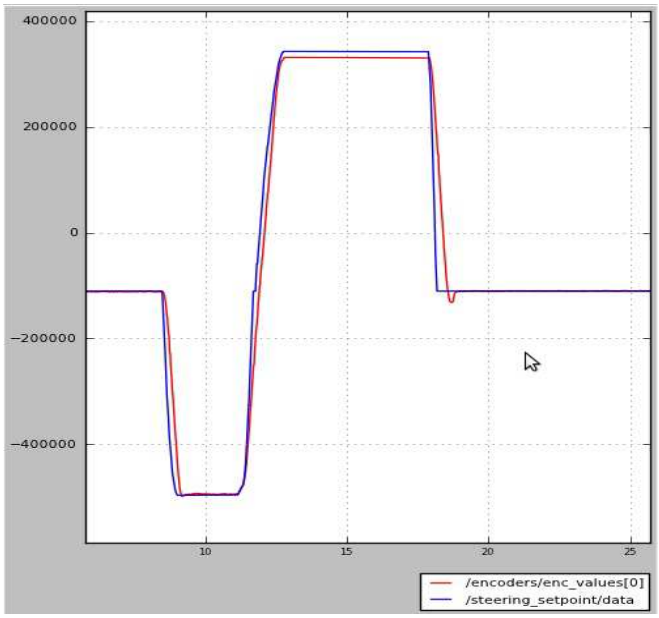


Figure 3-9: Command (blue) and response (red) of the steering PD controller

PD controller can be represented in time domain as:

$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t) \tag{3.4}$$

where,

$u(t)$: $Controller\ output$

$K_p$: $Proportional\ gain$

$K_d$: $Derivative\ gain$

$e(t)$: $Error\ between\ the\ setpoint\ and\ the\ current\ value$

Optimization of the controller parameters is not in the scope of this thesis. Therefore practical methods are used to tune the controller. First both proportional and derivative gains are set to zero. Proportional gain is increased to a point where maximum overshoot occurs with a reasonable rise time. At this point derivative gain is increased to damp the overshoot and oscillations. The resulted controller gains are as follows:

$K_p = 0.0025$

$K_d = 0.0003$

The response of the steering system to a set of steering commands given using the steering wheel is plotted in Figure 3-9. It is seen that the response follows the command with a small rise time and without an overshoot. This controller design is accepted for the application on the UGV.

### 3.4 Brake System

The brake system of the stock ATV is maintained and transferred on to the developed UGV. This brake system is composed of two subsystems. A hydraulic brake system is implemented at the rear wheels while front wheels have a wired system (Figure 3-10). A single cylinder block is used to pressurize the hydraulic brakes and a single wire is used for braking with the front calipers.

For the actuation of the brake systems, several alternative methods are investigated that are present in the literature. As seen from Table 2-1, primary methods used for brake actuation are hydraulic actuators, linear electric motors, DC and servo motors. Among these choices electrical linear actuator has not been taken into consideration

due to the unsatisfactory experience of Team Caltech [13] in the DARPA Grand Challenge. Due to its weight and maintenance problems a hydraulic system is also disregarded. Also, pneumatic system has been opted out in order to minimize the additional hardware on the robot.

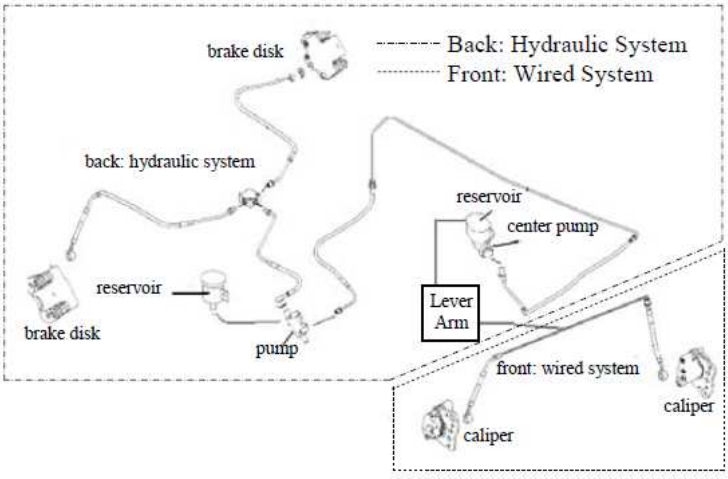


Figure 3-10: Two brake systems implemented on the vehicle

Schematic view of the designed brake actuation system is shown in Figure 3-11. The control of the pressure of the hydraulic cylinder and the tension of wired system are achieved by adjusting the position of the lever arm that is hinged at two points; one to the body of the vehicle and the other to the sliding table. The lever arm is driven by a DC motor through a ball screw assembly with rail way guidance. The ball screw has 3 mm pitch is connected to the DC motor by using a flexible servo coupler to compensate any dislocations. Position of the lever arm is limited by using two limit switches located at both extremes. Two inductance type proximity sensors are used as limit switches like the ones implemented in the steering system (Figure 3-12). A heavy duty permanent magnet geared DC motor having the specifications of 24 V DC nominal voltages, 1:15 gear ratio, and reversible rotation capability is chosen as the brake actuator. The motor is rated as 60 W. The motor has about 45 kg.cm continuous torque at the gearbox output while it rotates at 200 RPM which is sufficient to brake the vehicle.

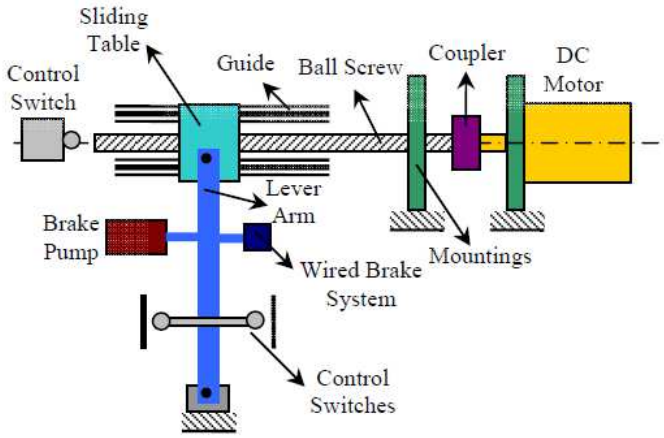


Figure 3-11: Schematic view of the brake actuation system

For the feedback purpose SIEMENS 6FX2001-3CC50 2500 PPR quadrature incremental encoder that is used in the steering system is mounted at the back shaft of the DC motor. With the use of the encoder the brake system is made controllable in a continuous manner instead of on/off braking strategy.

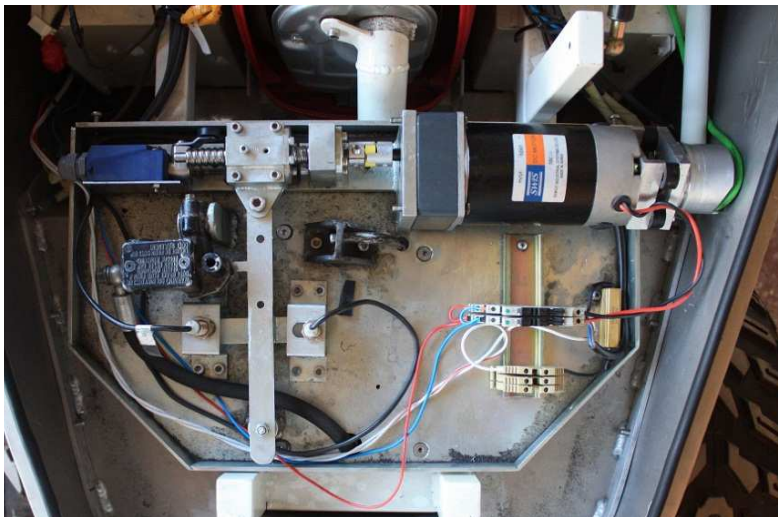

Figure 3-12: Developed brake system of the UGV

### 3.4.1 Brake Controller

Block diagram of the brake system is given in Figure 3-13. In this brake system, the desired brake percentage (from 0% to 100%) is commanded to the control computer via an interface device. A brake pedal is used as the interface in this study. Position of the brake motor shaft is measured with the industrial grade quadrature encoder whose pulse per revolution is 2500 which gives considerably high resolution. The encoder value is read via the data acquisition (DAQ) board mounted on the computer system. This signal is compared with the desired value and the error is supplied to a generic PD controller. The command value obtained from the PD controller is converted back to analog voltage again with DAQ card.

This voltage is fed to the DPRALTE-020B080 motor driver like the one in the steering system and the driver adjusts the output voltage value that will be sent to the brake motor to position it with the desired brake value.

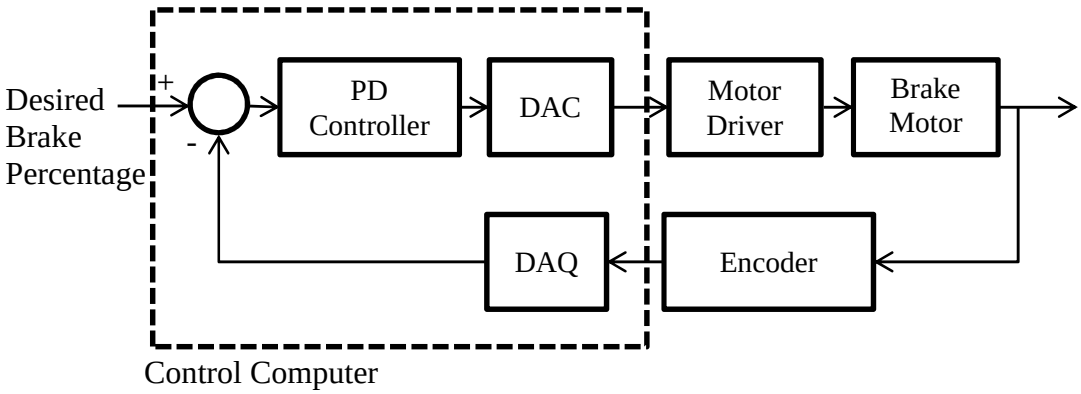


Figure 3-13: Block diagram of the brake controller of the UGV

From many available choices, a PD controller is designed to control the brake system like the controller implemented in steering system. The same controller which is given in (3.1) is used. It is tuned with the same method that is used in the tuning process of the steering controller.

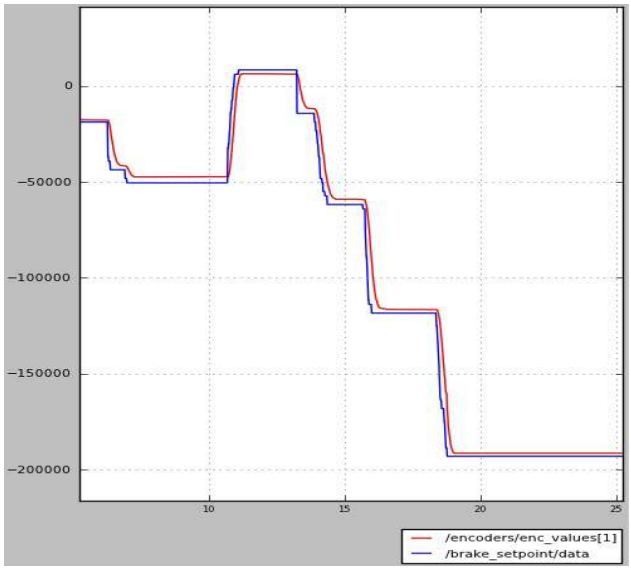


Figure 3-14: Command (blue) and response (red) of the brake PD controller

The resulted controller gains are as follows:

$K_p = 0.0005$

$K_d = 0.0001$

The response of the brake system to a set of brake commands given using the brake pedal is plotted in Figure 3-14. It is seen that the response follows the command with a small rise time and without an overshoot. There is a negligible steady state error. So there is no need to implement a PID controller. Thus this controller design is accepted for the application on the UGV.

### 3.5 Throttling System

Throttling system is inherited from the original stock vehicle. Rear axle of the ATV is mounted on the newly designed chassis of the UGV. Throttling is achieved using the two brushless hub motors mounted in the rear wheels (Figure 3-15). These 3-phase motors have a pancake motor structure. Each motor is rated nominally as 3000 W. Their nominal voltage is 60 V. At this nominal voltage, the motor has 860 RPM

rated speed, 47 A rated current and 34 Nm rated torque. When these motors are mounted in the wheels, UGV can have a maximum torque of 90 Nm or maximum speed of 70 km/h.

Figure 3-15: In-wheel motor that is mounted in the rear wheels

The motors have an efficiency value of 91% which is important for operation time of the vehicle. Motor has its own brake disc mounted on. A caliper is arranged on the axle according to this brake disc (Figure 3-16). Each motor weighs about 10 kg including the brake discs.

Figure 3-16: Hub motor, wheel, brake system, suspension and rear axle assembly

To be able to estimate the position and the velocity of the UGV a feedback system is needed at the wheels. For this purpose, an encoder disc is manufactured which has 64 notches on it (Figure 3-17). To count these notches and inductance type proximity sensor is located at 4 mm away from the encoder disc. In each full rotation of the wheel, the proximity sensor pulses 64 times which is a sufficient value for proper velocity estimation. These encoder discs are manufactured in two parts as semi circles which allows one to change them according to the needs of the research at a later time without disassembling the whole axle.

Figure 3-17: Proximity sensor and the encoder disc at the rear wheel

Off-road tires have been used for the designed vehicle. The dimensions of the front and rear wheels are as follows: radius and width of the front (steering) wheels are 28.5 cm and 16.5 cm, respectively. Those of the rear wheels are 26 cm and 25 cm, respectively. Front and rear tires' pressures are all set to 6.5 PSI.

### 3.5.1 Throttle Controller

The hub motors located in the rear wheels are driven via a 3000 W brushless motor driver (Figure 3-18). Each motor has its own driver. So they can be driven separately

even at different speeds. But the sake of the simplicity of the vehicle model, they are both driven with the same input command.

Figure 3-18: Hub motor driver (left). Drivers located next to the generator (right)

Working voltage of the motor driver is 60 V. It can drive the motor in both forward and reverse directions. A relay is mounted to set the direction of the drive via an interface of the driver. This driver has low voltage protection. When the supply voltage drops below 55 V, the driver halts the motor to prevent the battery system to be depleted. Also it has high temperature protection that is activated when the temperature of the driver board increases beyond 80°C. Another protection that is implemented in the driver is that, when the motor is blocked for 3 seconds without a movement, the driver stops the motor to prevent it from high current and possible hardware failures related to it.

The motor driver accepts drive commands via an analog voltage input. Voltage supplied to the motor is determined via the voltage division in the analog input leads of the driver. It accepts a voltage range from 0 V to 5 V but the active range is smaller than this. At no load condition, when the input voltage increases beyond 1.3 V the motor begins spinning. It reaches its maximum speed when the input voltage reaches 4.2 V. In between these limits, motor speed and the input voltage are directly proportional.

The speed estimation of the wheels is achieved via the encoder discs mounted in front of the proximity sensors. Pulses generated by the sensors are counted and the time elapsed between two pulses are used to calculate the velocity of the vehicle. With this feedback information, speed control of the vehicle is performed with the implemented PID controller. Block diagram of the throttle control system is given in Figure 3-19.

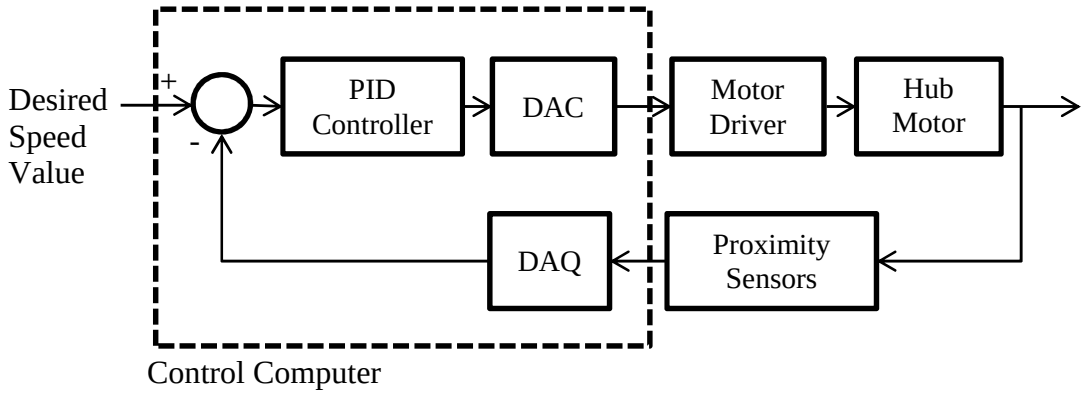


Figure 3-19: Block diagram of the throttle controller of the UGV

The desired speed value and the direction is indicated to the controller. Current speed is measured via the feedback system. These two values are compared and an error value is fed to the PID controller. Mathematical representation of a typical PID controller is given as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \qquad (3.5)$$

where $u(t)$: $Controller\ output$

$K_p$: $Proportional\ gain$

$K_i$: $Integral\ gain$

$K_d$: $Derivative\ gain$

$e(t)$: $Error\ between\ the\ setpoint\ and\ the\ current\ value$

$\tau$: $Integration\ variable$

The controller generates the necessary driver command and sends it to the DAC portion of the data acquisition board. DAQ board converts this signal to the analog voltage and sends this analog voltage to the motor driver. According to the analog voltage input, the driver adjusts the voltage that is supplied to the motor. Thus, a closed-loop velocity control is achieved.

### 3.6 Power System

Since the original ATV has no internal combustion engine and alternator all the energy demand of the vehicle and the hardware mounted on it is supplied with a high capacity battery pack. The capacity of the battery pack should be as high as possible to promise long operation time. However there is a trade-off between the battery capacity and the weight it adds to the total weight of the vehicle. The capacity value should be decided regarding these constraints.

Nominal voltage of the battery pack should be decided considering the hardware exists on the vehicle. For the sake of efficiency and simplicity of the power diagram of the vehicle, nominal voltage of the pack has to be as close as possible to the rated hub motor voltage which is 60 V in this case. With a selection of 60 V as the nominal voltage, the power needed by the motors can be directly supplied by the battery pack without any conversion or regulation. By this way, high current values that are required at the startup of the motors can be supplied easily.


Figure 3-20: The battery used in the battery pack. Greensaver SP20-12

Designed battery pack consists of 10 Greensaver SP20-12 silicone power batteries which have nominal voltage of 12 V and capacity of 20 Ah at 2 Hours rating (Figure 3-20). To supply the necessary 60 V, total of 10 batteries are divided into two groups of 5. Each 5 batteries are connected in series to obtain 60 V while these two groups are connected in parallel to double the capacity rating, namely 40 Ah (Figure 3-21).

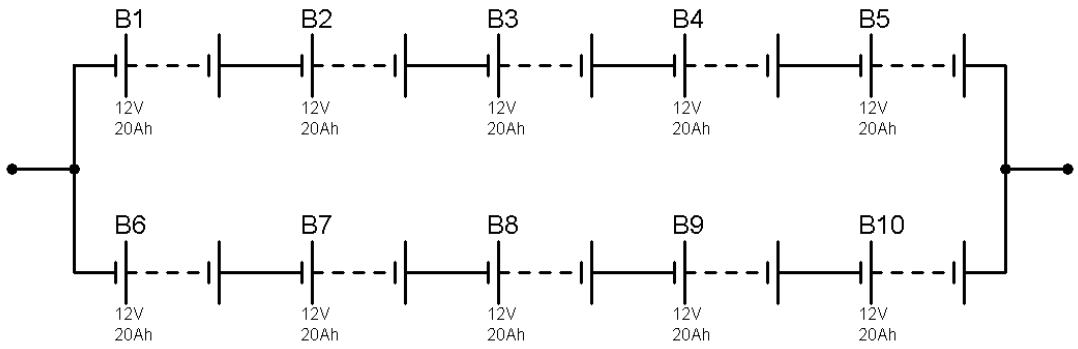


Figure 3-21: Battery pack circuit of the UGV

Each battery unit has a weight of 7.1 kg which results in 71 kg for 10 of them. Total weight of the vehicle is 320 kg which indicates that about %22 of the total weight is the weight of the main energy source, the battery pack. Increasing the energy capacity further will increase this percentage resulting in decrease of performance after a certain point.

Since the UGV is designed for outdoor and remote missions, lack of energy at any stage of the mission may become a major problem. To overcome this problem and to extend the operating range (away from the base) of the vehicle an auxiliary power system is designed and implemented. This auxiliary power system consists of a gasoline driven generator which has a maximum power rating of 1000 W at 120 V (Figure 2-8). Also it has a DC power output which supplies 8 A at 12 V. Its weight is about 14 kg which makes the total power system weight 85 kg. Generator only works if the battery pack is discharged and its voltage level is dropped below 55 V which is

the limit for low voltage protection of hub motor drivers resulting in the immobility of the UGV.

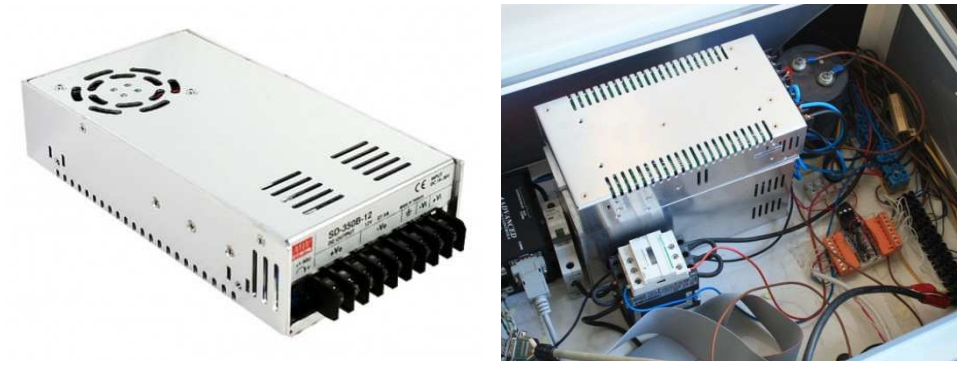

Figure 3-22: Meanwell DC-DC converter used (left) located in UGV (right)

To supply different required voltage levels to the corresponding hardware, three DC-DC converters are used (Figure 3-22). One converts from 60 V to 48 V with power output of 350 W while other converts from 60 V to 24 V with the same output power. A third voltage converter is implemented to convert from 60 V to 12 V at 500 W. All three converters that are used by various equipment are rated at different voltage levels.

### 3.7 Processing

UGV has many peripheral units and sensing equipment. Due to the large size of the vehicle and to achieve a reasonable operation speed and performance one or more powerful processing units are required. As listed in Table 2-3, there are various applications and choices that can be used as a processing infrastructure.

In this study, Intel i7 3.2 GHz CPU is selected as the main processing unit. This processor in mounted on a mainboard Gigabyte GA-H55M-S2H which has a form factor of ATX (Figure 3-23). Mainboard has two PCI-Express x16 and two PCI slots for the expansion boards. Back panel of the motherboard includes eight USB 2.0/1.1 ports which is important for external units to be connected. Mainboard has 8 GB of

DDR3 RAM with 1333 MHz bus speed is installed on the mainboard two provide necessary memory for the processes of the algorithms that will be run on the UGV (Figure 3-24).

Figure 3-23: Mainboard (left) and SSD (right) used on the vehicle

As the main storage unit a 32 GB solid state disc (SSD) is chosen. Since the vehicle is designed to be able to move in off-road conditions, SSD is the primary choice due to its performance in the conditions where large shocks and impacts are present. Also its data transfer rate of 3 GB/s is another factor for choosing SSD over conventional HDD units.

To be able to interface with the peripheral units that rely on RS-232 serial communication protocol, a two channel serial port board is mounted on the PCI slot of the mainboard.

Figure 3-24: Computer box with main control units (left) and regulators (right)

Since UGV has proximity sensors that are operate digitally, quadrature encoders that need an encoder interface, motor drivers that accept analog voltage inputs as command a data acquisition (DAQ) board is needed. For this purpose Humusoft MF624 multifunction I/O card is used (Figure 3-25). It is connected to the mainboard via PCI slot. The MF 624 contains 8 channel 14 bit A/D converter with simultaneous sample/hold circuit, 8 independent 14 bit D/A converters, 8 bit digital input port and 8 bit digital output port, 4 quadrature encoder inputs with single-ended or differential interface and 5 timers/counters. Its A/D converter has an input range of +/- 10 V. Proximity sensors are connected to the TTL compatible digital input ports of DAQ board. Encoders of steering and brake motor are wired to the encoder input and four analog outputs are connected to four motor drivers, two for in-wheel motors, one for steering motor and the other for brake motor.

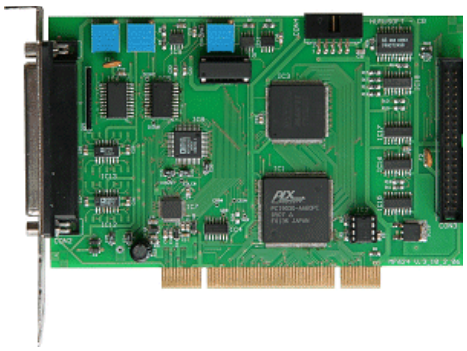

Figure 3-25: Humusoft MF624 data acquisition board

### 3.8 Environmental Sensing

It is seen in the Table 2-5 that using planar LIDAR and camera systems together is a common practice for sensing the environment. With the help of LIDAR systems upcoming obstacles that are laid in the planned path of the vehicle are gathered. The LIDAR systems are also used to roughly map the environment that the UGV moves in.

Two laser range finders are mounted on the vehicle for these purposes. One of the LIDARs in placed in front of the vehicle at 550 mm from ground and parallel to it (Figure 3-26). The second LIDAR system is placed at the back of the UGV to sense the environment and obstacle especially when moving backwards.


Figure 3-26: Laser range finder (left) mounted in front of the vehicle

The laser scanners are chosen as SICK LMS 291-S05 (Figure 3-26). LMS 291-S05 can generate scanning data at 75 Hz. Its scanning range can be programmed as 180° or 90° depending on the application. It can scan the region at 0.25, 0.50 or 1.00 increments allowing user to select the resolution. This laser scanner can measure a maximum range of 80 meters. But it is indicated in the features of the device that maximum range with 10% reflectivity is 30 meters. It is designed to operate in outdoor environments having IP 65 enclosure rating. Each scanner weights 4.5 kg, adding a total of 9 kg to the vehicle weight. They are connected to the control computer via RS-232 or RS-422 serial ports. For higher baud rates (up to 500K) RS-422 interface should be used.

For better mapping of the environment and the terrain, another LIDAR system, namely SICK LD-LRS 1000, is implemented on the vehicle (Figure 3-27). LD-LRS 1000 has a rotating mechanism at the top allowing it to scan a full range of 360 degrees. It provides a scanning frequency of 5 Hz to 10 Hz. It can be configured to take the measurement in increments of 0.062, 0.125, 0.25, 0.5, 1.0 or 1.5 degrees

depending on the resolution requirement of the project. It can measure from 0.5 meters up to 250 meters but the maximum range with 10% reflectivity is given as 80 meters in the specifications of the device.


Figure 3-27: SICK LD-LRS 1000 long range 360 degrees LIDAR

LD-LRS 1000 has three different communication interfaces. Serial interface which has a maximum baud rate of 115200 can be used in either RS-232 or RS-422 protocols. Ethernet interface provides communication speed up to 10 Mbit/s. The third interface is CAN bus which have a maximum data transmission rate of 1 Mbit/s. In this study Ethernet interface is selected due to its high data transmission rate and robust TCP/IP communication protocol. The scanner connected to control computer via onboard Ethernet port.

For depth perception purposes Bumblebee2 stereovision camera (Figure 2-11) of PointGrey is installed on the system. Bumblebee2 requires Firewire connection. Therefore a Firewire interface card is mounted on the PIC-E slot of the mainboard. Using this camera, depth information of the environment that the vehicle moves in is gathered. Also due to its stereovision capability, depth enhanced video stream of the environment is fed to the operator of the UGV.

As the driving camera of the UGV, Microsoft LifeCam Cinema 720p USB camera is placed in front of the vehicle. This camera can supply video with 720p HD resolution at 30 frames per second.

Microsoft LifeCam has a viewing angle of 74 degrees. This wide viewing angle is important for the remote operator. The effects of field of view (FOV) of the camera are investigated in the literature. The results summarized in Table A-3 suggest that wider FOV increases situational awareness and increases operator performance.

### 3.9 State Sensing

Throughout an assigned task the state of the UGV should be known both by the human operator and the control system implemented on the vehicle. State of the vehicle in 3-D world space is important to the operator for the sake of situational awareness. Also it is necessary for the localization nodes to predict the location of the vehicle in the local frame. A Crossbow IMU700 inertial measurement unit (IMU) is chosen as the primary state sensor of the vehicle (Figure 3-28).


Figure 3-28: Crossbow IMU 700 Inertial Measurement Unit

The IMU700 is a six-degree-of freedom (6DOF) Inertial Measurement Unit that provides monitoring of linear acceleration and angular rate. The IMU700 uses fiber optic rate gyro (FOG) technology. The sensor provides measurement of acceleration and rotation rate about three orthogonal axes. The three fiber optic rate gyros employ

the Sagnac effect [95] to measure angular rate independently of acceleration. The three accelerometers are silicon MEMS devices that use differential capacitance to sense acceleration. The IMU700 has three output options which are one analog and two digital modes. It is connected to the control computer via RS-232 serial interface. The sensory information gathered from three-axis accelerometer and three-axis gyroscope is used especially in the prediction phase of the Kalman filter that is used in localization algorithm which will be described in later sections of the thesis.

### 3.10 Localization

In outdoor applications, it is useful to know the localization of the UGV globally. In local frame, localization can be achieved with dead-reckoning fused with state estimation of inertial sensors. However localization with respect to the global frame is more challenging than to the local frame.

It can be concluded from the section 2.2.1.6 that the primary choice for localization in global frame is Global Positioning System (GPS) sensors. GPS sensors are not usable in crowded urban areas and tunnel-like road missions. However, especially for open terrain applications, GPS provides satisfactory results within an error range about 3-5 meters generally. But further improvement can be made on the GPS data with use of the method called Differential GPS (DGPS). In DGPS method, a ground station is deployed whose location is precisely known. A GPS receiver at this station calculates the error between its actual position and the position calculated based on GPS satellite data. This error value is broadcasted to nearly located GPS receiver and using this signal, GPS receivers correct their position. DGPS solution can provide localization results with less than 1-2 meters depending on the distance between the fixed and mobile units.

As a GPS receiver, Garmin GPS 10 Deluxe 12 channel GPS receiver is used (Figure 3-29). This device provides Bluetooth connection so there is no wiring required.


Figure 3-29: Garmin GPS 10 Deluxe GPS receiver

A DGPS system can be developed to have better position estimates. Alternatively, an ongoing project, European Geostationary Navigation Overlay Service (EGNOS), can be used. EGNOS is a service that collects satellite information from Ranging and Integrity Monitoring Stations (RIMS) that are located mostly across Europe, calculates correction data in Mission Control Centers (MCC) and sends this information to three geostationary satellites to be broadcasted back to the GPS receivers around Europe. With this correction signal, GPS receivers can estimate their position better (i.e. below 1-2 meters) if they are in the range of EGNOS project.

### 3.11 Communications

Teleoperated vehicles need a reliable line of communication between the vehicle and the remote operator. Also partly or fully autonomous robots need this communication to make the control center monitor vehicle movements and decisions on an assigned task. Lack of robustness in communication causes time delays and low update rates. These negative effects decrease operation performance of unmanned vehicles even can result in fatal errors for both UGV and the mission. A network connection relied on TCP/IP protocol is chosen as the communication tool between the UGV and the command center. Main reasons of this selection are reliability and wide bandwidth of TCP/IP network connection. Also, integrating other possible agents to the communication system becomes fairly straightforward by this method. AirTies WOB-201 54Mbps Wireless Outdoor model access point is used for the communication (Figure 3-30).


Figure 3-30: AirTies WOB-201 Wireless outdoor access point

WOB-201 has wireless data transfer rate up to 54 Mbps. It is designed for outdoor usage which makes it suitable for this study. It has Power over Ethernet (PoE) support, making no need to separate power cord. Its long range, which is up to 4 km for outdoor point-to-point connections, is important for a teleoperated vehicle project. It can operate in three different modes which are Bridge mode, Repeater mode, Access Point mode. In this study one WOB-201 is fixed in the ground station while a second one is mounted on the UGV. It is operated at 48 V voltage level. So the necessary voltage conversion done in the UGV from 60 V to 48 V via Meanwell SD-350D-48 DC-DC converter. A local area network (LAN) is constructed between these two devices. Command computer also joins this LAN, so it can reach the resources of UGV and send it command via this network.

### 3.12 Software and Control Architecture

Hardware subsystems of the UGV is implemented on the vehicle and the vehicle is made to be able to driven by wire. After these steps, for further applications and research studies a software framework and control architecture are needed. This framework should have features like modularity, robustness, ease of communication, wide library source and rich hardware support.

In the first stage, Microsoft Windows is chosen as a platform for software development. Gentle learning curve of C# and .NET Framework are the primary

reasons for this choice. A control software is developed using C# programming language and run on the computer that is mounted on the UGV which has Windows 7 operating system installed in. The need for developing each software from the scratch for each hardware for a main drawback of this self-developed software framework. Also, massive load of unused services and applications of Windows OS, deviated the software from even being close to a real-time system. Because of these drawbacks, eventually the interest on Windows OS and .NET Framework changed on to open source platforms such as Linux distributions.

As summerized in 2.2.2, there are some developed open source projects for robotic applications. Within this projects, ROS is selected for this unmanned ground vehicle application. Instead of Windows 7 platform, Ubuntu 11.04 (Natty Narwhal) which is a popular Linux distribution is installed in the UGV computer and further software development is carried on this operating system.

### 3.12.1 Robot Operating System (ROS)

Robot Operating System (ROS) is an open source software framework which is developed for robotic applications, providing operating system-like functionality on a computer network. ROS project is emerged during the collaboration between the STAIR project at Stanford and the Personal Robots Program at Willow Garage in 2007. Willow Garage is a robotics research company located in USA. They develop ROS since 2008. ROS includes primary libraries which are used commonly, hardware abstraction layers especially for widely used sensors and other hardware. It also has debugging tools that makes easy to visualize the architecture and find out bugs in the system.

Design goals for ROS was set as being peer-to-peer, tools-based, multi-lingunal, thin, and free and open-source [96]. With peer-to-peer topology implemented in ROS, processes which can be on different hosts are communicated directly with each other without a need to communicate over a central server. Instead, they need a master process functioning like a name server to find each other at runtime.

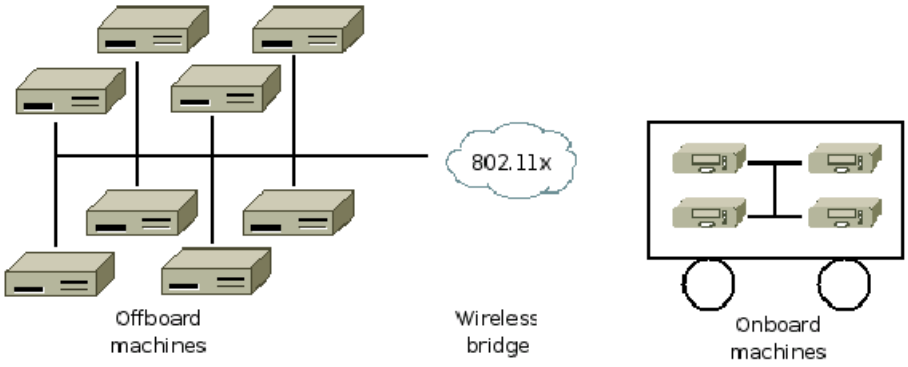


Figure 3-31: A typical ROS network configuration [96]

ROS supports multi-lingunality which allows a software developer to write a program in any programming language provided that a client library is ready also. ROS inherently has client libraries for programming languages C++, Python, Octave and LISP. ROS has various tools perform various tasks, e.g., navigate the source code tree, get and set configuration parameters, visualize the peer-to-peer connection topology, measure bandwidth utilization, graphically plot message data, auto-generate documentation, and so on. ROS re-uses code from numerous other open-source projects, such as the drivers, navigation system, and other algorithms. In each case, ROS is used only to expose various configuration options and to route data into and out of the respective software which makes ROS a “thin” framework. The full source code of ROS is publicly available and distributed under the terms of the BSD license, which allows the development of both non-commercial and commercial projects.

#### 3.12.1.1 Nomenclature in ROS

ROS has its own nomenclature for its components and services. When a ROS instance is run, before all other processes a process named as roscore is fired. Roscore is a master process which manages the addresses of other processes in the ROS network and manages messaging between them. Each process which is also an executable is called a node in ROS nomenclature. Nodes communicate with each other with help of roscore.

ROS has two messaging methods between nodes:

- Request-response method
- Subscribe-publish method

In the request-response method a process (service) waits for a request from another node and responses accordingly. In the time beginning from receiving request up to sending response, the node providing the service cannot communicate with other nodes. In the second method, subscribe-publish, a node (publisher) publishes messages, which have a defined type, to a certain topic which has a unique name, without considering whether there is a node listening (subscribed) to that topic or not. Node can publish a message at will or periodically at a defined rate. On the other side, a node (subscriber) listens a topic without considering whether there is a node publishing to that topic or not. In this method, nodes have no lock, which is present in the first method.

Nodes are connected to each other via topics and services. This whole topology constructed by the researcher using ROS is named as graph. At any instance of runtime, the interconnections can be viewed using the ROS tool, rxgraph.

Folders including reusable libraries are named as packages in ROS. Packages are distributable using repositories around the world. When packages are brought together they form stacks. Packages that have complementary functions can be collected under stacks for better grouping of files.

#### 3.12.1.2 Implementation

ROS is fully supported in Ubuntu, the most popular Linux distribution. Besides, it has experimental support in other Linux distributions and Unix-like operating systems like Fedora, Debian, Gentoo, OpenSUSE, Arch Linux. Also it has experimental support in Windows and OS X. However, platforms which are not open source like Windows and OS X are not targeted by the developers of ROS.

There is a large list of robots that use ROS or can be used with ROS [97]. Besides the robot platforms that are developed by Willow Garage (i.e., PR2, TurtleBot and Texai) many popular robotic agents can be used with ROS. ROS has many packages already that make it support wide range of hardware. Packages for most commonly used laser range finders, inertial sensors, cameras, etc. are readily available with their sources in ROS website, ros.org.

### 3.12.2 Control Architecture

To make use of sensors and other hardware implemented on the vehicle and to control the vehicle in a fully teleoperated manner, a control architecture is designed and related software is developed on ROS. This architecture is designed to be as modular as it can be to allow further hardware and algorithm implementations.

On the 32GB SSD mounted to the UGV computer, Ubuntu 11.04 (Natty Narwhal) is installed. On top of this Ubuntu distribution, ROS diamondback distribution is installed. On the other side, same configuration is setup on the remote control computer which is a notebook that has Intel i5 processor and 4GB of RAM. These two computers are linked together with means of wireless communication line whose construction is describer in section 3.11.

#### 3.12.2.1 Input Devices Node

Since the software architecture is designed to fully teleoperate the UGV, an input node is written to be able to send motion commands to the vehicle. As the primary input device steering wheel with brake and throttle pedals is selected (Figure 3-32).

Figure 3-32: Logitech Driving Force GT

This device is connected to the remote computer via USB port. Ubuntu has generic joystick support inherently but since this device has force feedback capability, it should also be supported. During research, it is seen that Ubuntu kernels before 3.2.0 do not have force feedback support in native mode for this Logitech device. Because of this reason, kernel of Natty Narwhal (ver. 2.6.38) is updated to version 3.2.0 to have full feedback support.

Input node, which is named as “joy_node” in developed software takes steering, brake and throttle commands through potentiometers as analog signal. Also, buttons on the wheel is taken as digital inputs for configuration during driving. Range of steering wheel is mapped to a range of +/- 100% scale. In its neutral position, steering command generated is 0%, while the limit in clockwise direction it is 100% and in reverse limit it is -100%. Similarly, full throttle and full braking are mapped as 100% while their neutral position is 0%. Finally the stick shift next to steering wheel is used for determinationof vehicle movement direction. Pushing the stick forward makes the UGV move forward while backward is the reverse. Any input device that can be mapped to these commands can be used to drive the UGV remotely.

joy_node publishes the input commands from a joystick or steering wheel as mapped to percentage values to a topic named “/joy”. Message type for this topic is defined in jostick package in “Joystick.msg”.

#### 3.12.2.2 LIDAR Node

Since laser range finders are mounted on the vehicle as described in section 3.8, LIDAR node is implemented in the software architecture. For SICK LMS 291 devices, “sicktoolbox_wrapper” package is used which is a part of “laser_drivers” stack maintaned by Chad Rockey [98].

Two parameters are fed to the sicklms node which connects to LMS 291 via serial port. One of the parameters is serial port name, and the second one is the baudrate of the serial connection. Sicklms node publishes laser range finder readings to “/scan”

topic. Message type for publishing to “/scan” is defined in “sensor_msgs” package with the name “LaserScan.msg”.

#### 3.12.2.3 DAQ Node

Since Humusoft MF624 data acquisition I/O board is installed on the UGV computer, a suitable driver that will work in Linux environment is needed. There is no official library distributed by Humusoft targeting Linux distributions. So a third party library, hudaqlib, is obtained written for Linux [99].

A node is written with the name “mf624_node” to control DAQ interface using the hudaqlib library. This node reads all analog input ports and publishes them in the topic named “/AIs”, digital inputs in “/DIs” and encoder values in “/encoders” topics with related message types developed with the node. Each reading and publish is done in 1 ms, i.e., this node operates in 1 kHz.

Commands to the external hardware are also extracted via this node. It subscribes to motion command topics and according to the published messages to “/DOs” and “/AOs” topics mf624_node generates outputs from digital and analog output ports respectively.

#### 3.12.2.4 Speed Node

Speed node which is named as “speed_node” in the architecture is responsible for measuring and monitoring speeds obtained in the wheels and steering angle. Using these values, speed node generates odometry value and publishes it to the topic “/odom” for further autonomy and feedback studies.

Speed node measures wheel speeds by subscribing to “/DIs” topic and gathering digital input information published by “mf624_node” node. Inside digital input messages, speed_node takes the ones that are belong to rear wheel proximity sensors which are placed next to encoder disk for speed measuring purposes. Speed node counts the number of notches that are observed and converts this value to position

information. Also takes the derivative of this value to estimate the velocities of the rear wheels.

#### 3.12.2.5 Arbitrator Node

Arbitrator node is responsible to determine the highest priority behavior and generate commands according to it. This node subscribes to various command and behaviour sources. It subscribes to “/joy” topic to gather direct movement commands from teleoperator, to “/scan” topic to get near obstacle information to prevent the UGV from crashing to objects, and to “/heartbeat” topic to listen for heartbeat from remote control to stop the vehicle if the communication breaks.

Arbitrator takes “/heartbeat” topic into consideration in the first place. If heartbeat is not satistified, arbitrator generates motion commands to stop the vehicle immediately disregarding the commands taken from “/joy” topic. If heartbeat is present but “/scan” topic reports a nearby object that will cause the UGV to crash, again arbitrator disregards “/joy” commands and it does not allow the teleoperator to drive the vehicle in the dangerous direction. Finally if both “/heartbeat” and “/scan” do not report emergency, arbitrator takes direct input commands from “/joy” topic and generates motion commands to be sent to mobility nodes.

#### 3.12.2.6 Motion Decomposer Node

Motion commands that are generated by arbitrator node is gathered by “motion_decomposer” node. The task of this node is divide the motion commands to related nodes that control the mobility of the UGV. There are three nodes that subscribe to “/motion_commands” topic in which motion decomposer node publishes.

Motion commands generated by this node are still in the form of percentages for steering, throttle and brake. Besides, they carry a binary direction information which is -1 for reverse and 1 for forward movement.

#### 3.12.2.7 Steering Node

Steering node is responsible to take motion commands from motion decomposer node and convert it to necessary analog voltage commands to steer the front wheels to the desired angle. It publishes the required analog output value to "/AOs" topic which is listened by mf624_node data acquisition board control node. As mentioned in section 3.3, an incremental quadrature encoder is installed to the steering motor to control its position. Therefore, at the first power up of the system, steering_node drives the steering motor in clockwise direction until it reaches to limit sensor. At this point it records the encoder reading and starts to drive motor in the opposite direction until the limit sensor at the other side. When it reaches to the limiting sensor, it again records the encoder reading. Then the node scales this encoder readings range to +/- 30 degrees of steering angle. Further control of steering wheels is maintained with this reference values obtained in the start up.

```
1: while limit_values not found
2:    while right_lim not found
3:          drive steering_motor to right
4:    end
5:    record right_lim
6:    while left_lim not found
7:          drive steering_motor to left
8:    end
9:    record left_lim
10:   encoder_range = right_lim - left_lim
11:   set limit_values to found
12: end

13: start:
14:   get st_command
15:   get st_feedback
16:   setpoint = (right_lim - left_lim) * (st_command)
17:   error = setpoint - st_feedback
18:   deriv = (error - previous_error)/dt
19:   output = (Kp*error) + (Kd*deriv)
20:   previous_error = error
21: goto start
```

Figure 3-33: Pseudocode written for steering_node

Pseudocode for the steering_node is given in Figure 3-33. In the first segment from line number 1 to 12, node looks for limiting values in both directions. When these limits are found it operates in normal PD controller mode. PD controller is implemented in “start” procedure, or from line number 13 to 21.

#### 3.12.2.8 Brake Node

Brake mode operates identically to steering node. In the first power up it looks for limits of braking mechanism and converts the brake command into analog voltage command according to these found limits. Therefore “brake_node” is not given in detail. Please refer to section 3.12.2.7 for identical application and pseudocode for steering controller.

#### 3.12.2.9 Thrust Node

Thrust node is the node which is resposible for controlling the hub motors located in the rear wheels. Since these motors are the main traction providers “thrust_node” is the node that controls the main mobility of the UGV.

Thrust node maps the commands taken from the throttle pedal of the input device to analog output voltages that will command the hub motor drivers and eventually make the motors spin. It can operate in both closed loop PID and open loop proportional speed controller modes. In close-loop PID speed controller mode, thrust_node gets the speed information from speed_node and uses this value to control the speed of the vehicle using a PID controller. On the other side, open-loop proportional speed controller only maps the command to analog voltage output and does not check for state of speed of the UGV.

### 3.12.3 Bringing Nodes Together

Since ROS is based on TCP/IP network protocol, it can operate on multiple machines that are present in the same network. In this type of application of ROS, only one master node (roscore) is run and all nodes distributed along multiple machines connect to this master to make it able to record their addresses and configurations.

In the configuration implemented on the UGV, there are two computers: UGV computer and remote control computer. These two computers are connected to each other using the master node which is run in the UGV computer. UGV computer hosts all the nodes except joystick node which is hosted by remote computer since the input device is connected to it.

### 3.13 Conclusion

A development of a large-scale UGV from scratch using some preliminary parts of a commercially available ATV is explained in this section. Steering, brake and throttling systems are made available to be driven-by-wire. Power system is constructed according to the energy needs of the electronics on the vehicle. A control computer is installed with interface boards on it. For autonomy applications that will be developed, environmental sensors are located around the UGV. Inertial sensors and GPS localization sensors are mounted. A communication link is established between the vehicle and the command center via TCP/IP network connection with aids of long range wireless access points. Finally a software framework and control architecture are designed and/or implemented.

At this point of the study, the UGV is developed and made ready for further developments and research studies. After this development phase, study continued with force feedback application and its effects to workload of the operator.

# CHAPTER 4

# HUMAN-ROBOT INTERACTION AND WORKLOAD MANAGEMENT

## 4.1 Introduction

Teleoperation of unmanned ground vehicle is a challenging task for humans especially when the control interface is designed and applied improperly. The way auditive, visual and haptic feedbacks provided to the user completely affects the user performance. A proper application can improve teleoperation performance significantly while improper application can lead to totally unusable teleoperation interfaces that make teleoperation so challenging or even impossible.

Teleoperation is a task, in general, imposing high levels of workload on human operators who work within remote agent systems. For a better performance, workload on human operator must be decreased to a reasonable level. There are many research that are conducted with this purpose in the literature. Summaries of these research are listed in Table A-1 to Table A-10. These tables show that basic guidelines are formed in the literature that will help to manage workload on the operator throughout a remote operation task. These research studies mainly focused on how to supply visual feedback to the teleoperator (Table A-3 to Table A-6), effects of task criteria (Table A-7, Table A-8), how to decrease workload with introduction of autonomy (Table A-9, Table A-10) and effects of degraded communication lines to the teleoperation performance between the agent and the operator (Table A-1, Table A-2).

Within last few decades, development and use of hi-tech robotic equipment resulted in complex cognitive loads in human robot interactions. In these interactions, attention of the operator is divided between various display and operation systems.

Multimodal systems and technologies can provide the operator with a means of improving situational awareness (SA) and reducing workload when these technologies used to support conventional manual controls and visual displays.

### 4.2 Human-Robot Interaction (HRI)

In 1942, Isaac Asimov stated three laws of robotics in his short story “Runaround” [100] as follows:

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

Although these laws were stated in a science-fiction novel, it is well accepted and gained popularity by the community of researchers who study in HRI field. HRI is a research field mainly focused on how should modern robotic agents and human beings interact with each other efficiently and effectively. HRI field emerged and gained exploded popularity with new equipment and development of autonomy technologies in few past decades. HRI is a multidisciplinary field with contributions from human-computer interaction (HCI), robotics, human biology, psychology, sociology and similar disciplines. This study contributes to HRI literature mainly in field of view of robotics and HCI.

Controlling a platform or interacting with an artificial agent consists of many tasks. Examples include executing menu functions, navigating to waypoints, manipulating a foreign object, processing information from data links, communicating with team members, and in some cases, physically moving or interacting with the platform. These all tasks should be evaluated in term of HRI to reduce workload of human operator and increase teleoperation performance.

### 4.2.1 Principles

Throughout the HRI history there have been many attempts to define guidelines and principles for efficient interaction. Goodrich and Olsen [101] listed a set of seven principles for efficient human robot interaction. In their study, they defined two interaction loops as the interaction between human and robot via an interface, and interaction between robot and the external world via an autonomous control. Limitations on autonomous movement of robots and interaction channels restrict human intention by the available recent technology.

Goodrich and Olsen list five bases for their interaction principles. First basis is *neglect time*. It is a measure of time which defines the duration that robot can act autonomously without a need from a human. Next basis is *interaction time* which is the duration between the human intervention and the moment that robot reaches its maximum performance. The third basis is *robot attention demand*. Robot attention demand is the ratio of interaction time to sum of interaction time and neglect time. The fourth basis is *free time*. It is the measure of the time left to use to execute other tasks. The last one is *fan out*. Fan out is the number of robots that can be controlled simultaneously and effectively by a human. On top of these bases, seven principles are listed as:

1. *Implicit Switch Modes*: Switching between interfaces and modes of autonomy should not require much time and user effort. The user should know only how to act in each mode and interface.

2. *Use Natural Cues*: Robotic assets should be as natural as possible where the naturalness is defined as availability of calibrated mental models, and practiced short term memory.

3. *Directly Manipulate the World*: The user should not think about robot itself. He should have direct contact with the environment. For example, he should be able to navigate the robot to a destination point by only touching that point on the screen.

4. *Manipulate Robot-World Relationship*: The relation between the robot and the environment around it should be explicit to the user. This principle is especially for the situations where a direct connection between the environment and the user is not available.

5. *Information is Meant to be Manipulated*: The user should be able to change what is fed back to him. For example, if a speed information is provided to the user, he should be able to speed up or slow down the robot.

6. *Externalize Memory*: Short-term memory of the user should not be occupied by past sensory information or sensor fusion and integration models. This principle aims to decrease cognitive workload. For example, in a mobile robot teleoperation scene, the environment around the robot but outside the field of view of the drive camera should be supplied to the user and this information should be externalized from his short term memory.

7. *Support Attention Management*: A properly designed and implemented interface should direct the attention of the user to the critical points and directions. For example, an object which is critical for the task should change color or blink to gather attention.

These seven principles are main guidelines developed by Goodrich and Olsen to provide effective interfaces in human robot interaction context. They are widely accepted and implemented in various applications in HRI field.

### 4.2.2 Human Role

There have been many studies which define the human role in a human-robot interaction context. In [102], Schreckenghost states that the role of human can change according to the environmental conditions during a task. Human role is

divided into three main types in another study [103]. These roles are named as supervisor, operator and peer. Human as a supervisor role only intervenes or manipulates when the robot requires. Other than that, the supervisor only monitors the situation of the agent. As an operator, human controls the robots, perceives the sensory information and manipulates the actions according to inputs and task goals. Operator is also responsible of fixing malfunctions of the assets. Human as a peer generally take place in multiple teams consist of humans and robots. Murphy divided the roles of humans into two as operators and problem solvers [34]. In this categorization, operators control the robot motion while problem solvers control the missions and the data gathered by the robot from the environment.

## 4.3 Teleoperation

Teleoperation is the manipulation of the vehicle state remotely which is separated from its operator in terms of space and/or time (without being co-located or simultaneous). Teleoperation involves human and a remote asset in its context. Involvement level of the human operator to the control of the vehicle and the interaction between depends on the control system and autonomous aids provided to the operator. At the minimum level of these aids, operator controls the vehicle in a fully manual manner (pure teleoperation). If the autonomous aids are at their maximum, which is the condition that there is no need for human intervention, the full autonomy is present. Human performance issues are investigated in this section in teleoperation context.

### 4.3.1 Remote Perception

Gathering sensory information from a robotic agent that executes a mission in a remote environment through its sensors and other hardware channels is named as remote perception. For efficiency and success of teleoperation or telemanipulation missions, remote perception at a reasonable rate is essential. Since the operator is remotely located from the environment, the perception channels of robot and human do not overlap. For example, a visual feedback of acceleration taken from a camera is not completed with an acceleration sense in ear. Therefore poor perception results

in degraded situational awareness (SA). One of the negative effects of poor perception is scale ambiguity [104]. Also if remote camera is not placed in a proper height that matches with a human point of view, this can make teleoperation task harder for a human operator [105]. Especially for telemanipulation missions, estimation of size and position of remote objects to be manipulated (e.g., bomb setups) is crucial [106].

Steinfeld et al. developed common metrics for human robot interaction based tasks for standardization [107]. They divided perception into two subtasks as *passive perception* which is the interpretation of readily available sensor information, and *active perception* which is an act to seek new sensor information to enhance situational awareness. Remote perception is affected by time delays present in the control and/or communication systems, limited video image quality and viewpoint of the camera located on the remote agent etc.

### 4.3.2 Teleoperation User Interfaces

A proper user interface design and implementation is the main factor that affects the performance of a human operator during human-robot interaction missions. Teleoperation user interfaces should provide a few main features that can increase the SA of the operator, decrease his workload and improve the performance.

Pitch and roll motion of robotic assets are hard to estimate for an operator if a reference point, like horizon, is not available. Due to this limited remote perception, a robot may roll over without intention of its operator [48]. Therefore a display which includes attitude information is useful especially for stressful tasks and complex mission environments. To increase visual remote perception, stereoscopic displays are employed in teleoperation interfaces. They provide more accurate estimation of a remote object or environment and result in less task errors. But they can induce extra stress on the operator and cause motion sickness if they are not implemented properly [47].

Since teleoperation missions highly rely on visual perception, cognitive load on visual channel can be excessive for human. Therefore other modalities of human body for teleoperation should be used for efficient tasks with high performances. Next section is dedicated to multimodal user interfaces which employ other modalities of data channels.

### 4.4 Multimodal User Interfaces

Multimodality is defined as “the capacity of the system to communicate with a user along different types of communication channels and to extract and convey meaning automatically” [108]. Human body has different modalities (i.e., sensory channels) for gathering sense input from and extracting commands to the outer world. These modalities which can be listed as visual, auditive, haptic, olfactory, gustatory and vestibular modalities [109] are can be used as a feedback or command mechanism when interacting with a robot, or more generally a computer. Multimodal interaction makes use of more than one of these modalities in interaction with a computer system.

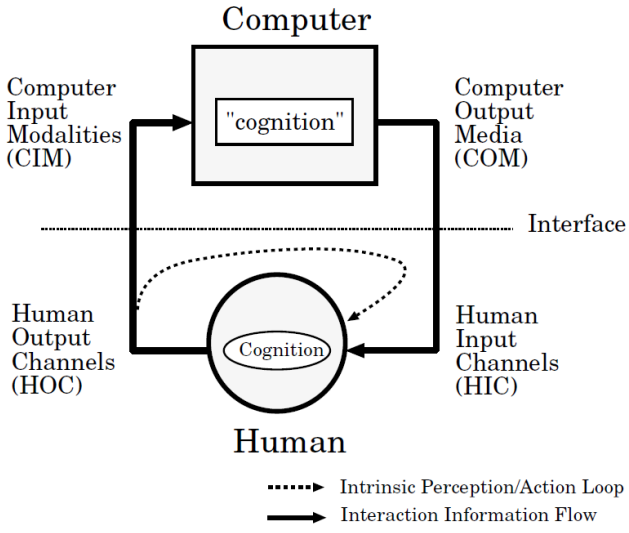


Figure 4-1: Multimodal human-computer interaction [110]

Benefits of multimodal interfaces are listed as [111]:

- Efficiency – For example, objects with geospatial extent are efficiently identified with a map and gesture; identifying sets of objects with abstract properties is done more quickly linguistically (i.e., using each modality for the task that it is best suited for).
- Redundancy – For example, conveying the location of an entity both in language and gesture can increase the likelihood of reference identification. (i.e., communication proceeds smoothly because there are many simultaneous references to the same issue).
- Perceptability – For example, certain tasks such as feature detection, orientation, and/or reference identification may be facilitated in spatial context.
- Naturalness – For example, empowering the user to use those forms of communication chosen in human-human interaction (i.e., human-computer communication that is close to human-human communication).
- Accuracy – For example, gesture may enable more precise spatial indication over voice (i.e., another modality can indicate an object more accurately than the main modality).
- Synergy – For example, one channel of communication can help refine imprecision, modify the meaning, and/or resolve ambiguities in another..

Silbernagel [109] summarizes sensory perception, related sense organ and modality in six categories (Table 4-1). Most dominant and commonly used modality of human sensation is visual modality. However parallel or sequential use of other modalities can decrease human cognitive load and increases human-computer interaction performance if they are designed and applied properly.

Table 4-1: Senses and modalities

| **Sensory Perception** | **Sense Organ** | **Modality** |
|---|---|---|
| Sense of sight | Eyes | Visual |
| Sense of hearing | Ears | Auditive |
| Sense of touch | Skin | Haptic |
| Sense of smell | Nose | Olfactory |
| Sense of taste | Tongue | Gustatory |
| Sense of balance | Organ of equilibrium | Vestibular |

Multimodal interaction and multisensory feedback can exert significant cognitive demands on users. Cognitive resource of human is limited. Attention to be diverted to an input is selective and has limited capacity. Considering the capacity of short-term memory of a human, taking into account the human ability when designing the interface, is essential for control and learning affordances.

### 4.4.1 Haptic and Force Feedback

The word “haptic” is originated from a Greek word “haptesthai” and has a meaning as “relating to or based on the sense of touch” [112]. Touch is one of the most primitive and pervasive sense of human beings that starts to develop in uterus beginning from week 8 of the gestational period [113]. The senses of touch are mediated by the somatosensory system in medicine. Somatosensory system is concerned with sensory information from the skin, joints, muscles and internal organs. Three main modalities are derived from somatosensory system:

- Discriminative touch (Tactile/Cutaneous)
- Temperature and pain (Tactile/Cutaneous)
- The kinesthetic senses (Proprioception)

In this study, kinesthetic modality (proprioception) is taken into consideration. Kinesthetic receptors are muscle and joint receptors (proprioceptors) which are located in tendons, muscles and joints. They perceive position and movement

information. Force feedback to the human operator in a teleoperation mission is received with these receptors.

Force feedback interfaces can be viewed as having two basic functions [114]. First function is to measure the positions and contact forces of the user's hand (and/or other body parts), and the second function is to display contact forces and positions to the user. In teleoperation basis, force feedback interface can be a steering wheel which has one degree-of-freedom or a joystick with two degrees-of-freedom to command the vehicle movement and gather feedback generated from this movement.

Haptic modality with use of skin and limbs is not well investigated in the literature in comparison to other modalities as visual and auditive. Use of haptic modality in conjunction with visual modality can increase the operator performance, especially in teleoperation tasks of unmanned vehicles. Chapter 5 is dedicated to the study on this topic.

# CHAPTER 5

# FORCE FEEDBACK TELEOPERATION WITH THE DEVELOPED UGV

## 5.1 Introduction

Although there are many research studies in the literature that address the methods which can be used to reduce workload in human operator during teleoperation missions, providing haptic feedbacks to the teleoperator and its effects on the workload and operation performance are not considered significantly. Since sense of touch is a strong channel of human body to get feedbacks from the world, it should be considered in decreasing workload in teleoperation tasks of remote agents. In this chapter, effects of force feedback from UGV to the teleoperator are investigated in an obstacle avoidance scenario.

In the first section, vehicle model, motion planning and obstacle avoidance methods used are described. Simultaneous localization and mapping (SLAM) algorithms that are implemented are given with the software developed for purpose of this chapter. In second section force feedback method and developed software are mentioned. Finally the experiment that is conducted to verify the method and its results are discussed.

## 5.2 Motion Planning and Navigation

To accomplish a given task, a robot should understand and interpret its surroundings using its sensors mounted on. Understanding the environment requires capturing of the sensory information at the right level of abstraction. To navigate to a given target point, robot should know the environment that it is in and also it should know or at least estimate where it is in this environment.

Although a teleoperated UGV is in the scope of this thesis, similar requirements are needed when path planning and obstacle avoidance features are integrated into teleoperation control mode. Difference is that, in the method proposed in this research study results of path planning and collision avoidance algorithms are subject to human teleoperator confirmation. In other words, path planning and obstacle avoidance algorithms suggest the teleoperator the commands that will navigate the UGV in the best path through the obstacle by the means of force feedback applied to the steering wheel. Though, the operator can overcome the force developed on the steering wheel and take totally different path according to his internal decision mechanisms. Aim of the proposed method is to reduce operator workload and increase teleoperation performance by assisting the operator through force feedback applied to steering wheel.

Because of these mentioned reasons a navigation capability should be added to the UGV like an autonomous robot. The difference will be that, the throttle command is only generated by the teleoperator and he can either leave the steering commands to the UGV itself or guide the vehicle any other direction he decides disregarding the internal algorithms of UGV.

A map of the environment is required for navigation. This map can be readily available prior to the operation or it can be generated dynamically during runtime. In this study, latter one is used to widen the operation capabilities of vehicle. The map is unknown prior to task and UGV builds map as it navigates.

### 5.2.1 Vehicle Model

Developing a satisfactory navigation system requires the vehicle model for position and velocity calculations. A kinematic lateral motion model of a vehicle can be developed under certain assumptions listed below. In such a model, forces acting on the vehicle body and its subparts are neglected and a kinematic mathematical model is derived. Since the forces are neglected, only geometric relations are taken into consideration while deriving the equations of motion.

A model of the vehicle which is named as bicycle model is used as shown in Figure 5-1 [115]. In the bicycle model, right and left front wheels are combined and represented as a single wheel at point A while right and left rear wheels are combined and represented as one wheel at point B. The model is derived in a general manner which assumes both front and rear wheels can be steered. For the case of this study, since only the front wheels can be steered, rear steering angle will be set to zero in resulting equations of motion.

In the Figure 5-1, front and rear steering angles are represented by $\delta_f$ and $\delta_r$ respectively. Point C represents the center of gravity of the vehicle.

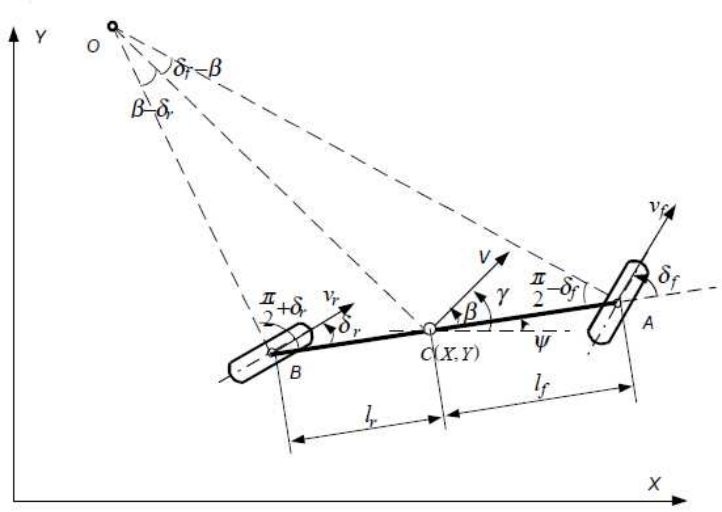


Figure 5-1: Kinematic model of a vehicle [115]

The representations $l_f$ and $l_r$ are the distances of points A and B from the center of gravity of the vehicle respectively resulting in definition of the wheelbase as: $L = l_f + l_r$. Three coordinates, $X$, $Y$ and $\Psi$ are present to describe the planar motion of the vehicle in inertial reference frame. $\Psi$ is named as heading angle while $\beta$ is the slip angle of the vehicle which is defined as the angle between velocity of the vehicle at center of gravity and the longitudinal axis of the vehicle.

Assumptions for the validity of the vehicle model are listed as:

1. Vehicle has a low-speed motion (i.e., less than 10 m/s).
2. Slip angles of front and rear wheels are assumed to be both zero.

If the vehicle is accepted to move at low speeds, the lateral force generated by the tires is relatively small. Moving on a curvature with radius of R generates total lateral force of

$$\frac{mV^2}{R}$$

which varies with the square of speed V and is small at low speeds. So both assumptions can be accepted for the development of this vehicle model. Calculations for equations of motion are as follows.

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin(\frac{\pi}{2} - \delta_f)}{R} \tag{5.1}$$

$$\frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin(\frac{\pi}{2} + \delta_r)}{R} \tag{5.2}$$

Multiplying both sides of Eq. (3.6) by $\frac{l_f}{\cos(\delta_f)}$ and Eq. (3.7) by $\frac{l_r}{\cos(\delta_r)}$ and adding the results,

$$\left[\tan(\delta_f) - \tan(\delta_r)\right]\cos(\beta) = \frac{l_f + l_r}{R} \tag{5.3}$$

Since rate of change of orientation of the vehicle is,

$$\dot{\Psi} = \frac{V}{R} \tag{5.4}$$

combining Eq. (5.3) and (5.4) result is

$$\dot{\Psi} = \frac{Vcos(\beta)}{l_f + l_r}\left[\tan(\delta_f) - \tan(\delta_r)\right] \tag{5.5}$$

And other two equations of motion are,

$$\dot{X} = Vcos(\Psi + \beta) \tag{5.6}$$

$$\dot{Y} = Vsin(\Psi + \beta) \tag{5.7}$$

Finally,

$$\beta = tan^{-1}\left(\frac{l_f \tan(\delta_r) + l_r \tan(\delta_f)}{l_f + l_r}\right) \quad (5.8)$$

In this generalized kinematic vehicle model, three inputs are present: front steering angle ($\delta_f$), rear steering angle ($\delta_r$) and velocity (V). However, the developed UGV can be steered by the front wheels only. Therefore, setting $\delta_r = 0$ results in the following final equations.

$$\dot{X} = Vcos(\Psi + \beta) \quad (5.9)$$

$$\dot{Y} = Vsin(\Psi + \beta) \quad (5.10)$$

$$\dot{\Psi} = \frac{Vcos(\beta)}{l_f + l_r}\tan(\delta_f) \quad (5.11)$$

$$\beta = tan^{-1}\left(\frac{l_r \tan(\delta_f)}{l_f + l_r}\right) \quad (5.12)$$

These equations are used in the navigation algorithms developed for the position and velocity calculation of UGV.

### 5.2.2 Simultaneous Localization and Mapping (SLAM)

Location of the robot should be known in mobile robotics tasks especially if collision avoidance is implemented. A robotic agent can move autonomously or plan a path only if it can localize itself in the given map of the environment. Localization problem is also named as position estimation.

Besides localization problem, if the prior map of the environment where the robot is exploring and localizing itself according to is not available or unknown then the problem becomes a hard one which now includes proper mapping of the environment. This problem is called “Simultaneous Localization and Mapping” which is commonly abbreviated as SLAM. SLAM is regarded as a chicken-or-egg problem by Thrun [116]. That is, a map is needed for localizing a robot while a good pose estimate is needed to build a map. To solve this problem a robot should move a

bit while acquiring a map of the environment and localizing itself within this generated map simultaneously, the procedure which gives the name of SLAM problem. As it moves further robotic agent should correct the map according to its new sensor readings.

SLAM problem has two main forms from a probabilistic perspective: Online SLAM problem, and Full SLAM problem. Online SLAM estimates most recent pose and map. It estimates the posterior over the instantaneous position in the map. Posterior is given as:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) \tag{5.13}$$

where $x_t$ is the position at time $t$, $m$ is the map, $z_{1:t}$ is the measurements from time zero to $t$ and $u_{1:t}$ is the controls similarly.

On the other side Full SLAM calculates the posterior over the full path $x_{1:t}$, instead of the most recent pose $x_t$ which is the issue in online SLAM. Full SLAM is the preferred method in this study. Resulting in a posterior defined as:

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \tag{5.14}$$

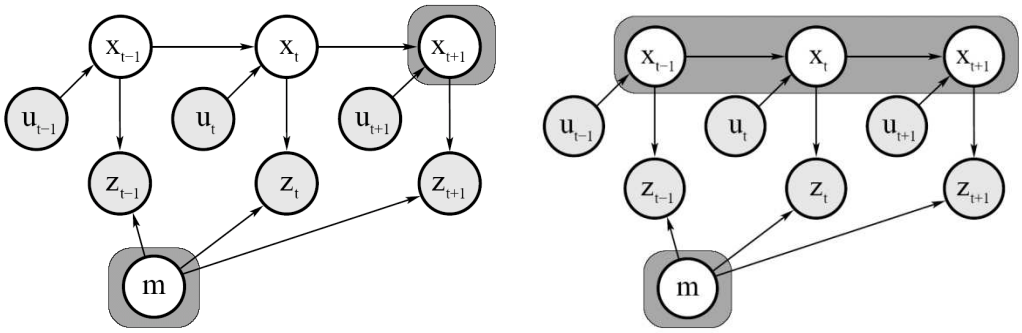


Figure 5-2: Bayesian representation of online (left) and full (right) SLAM [116]

There are various techniques to solve SLAM problem. These techniques can be listed as Scan Matching, Extended Kalman Filter (EKF) SLAM, FastSLAM, GraphSLAM, and Sparse Extended Information Filter (SEIF) SLAM [116]. FastSLAM is the algorithm that is implemented in this study.

#### 5.2.2.1 The FastSLAM Algorithm

FastSLAM is an efficient variant of SLAM algorithms. It divided SLAM into to independent problems as localization and landmark estimation [117]. FastSLAM implements an algorithm which includes the Rao-Blackwellized particle filter [118]. Rao-Blackwellized is a modified particle filter that can be applicable to nonlinear models that can estimate the posterior on robot paths. In this filter, each particle has K many Kalman filters that estimate the K many landmark locations. This implementation results in O(MK) computation time. M represents the number of particles while K defines the number of landmarks.

- Do the following *M* times
  - **Retrieval**. Retrieve a pose $x_{t-1}^{[k]}$ from the particle set $Y_{t-1}$.
  - **Prediction**. Sample a new pose $x_t^{[k]} \sim p(x_t | x_{t-1}^{[k]}, u_t)$.
  - **Measurement update**. For each observed feature $z_t^i$ identify the correspondence $j$ for the measurement $z_t^i$, and incorporate the measurement $z_t^i$ into the corresponding EKF, by updating the mean $\mu_{j,t}^{[k]}$ and covariance $\Sigma_{j,t}^{[k]}$.
  - **Importance weight**. Calculate the importance weight $w^{[k]}$ for the new particle.
- **Resampling**. Sample, with replacement, *M* particles, where each particle is sampled with a probability proportional to $w^{[k]}$.

Figure 5-3: The basic steps of the FastSLAM algorithm [116]

FastSLAM algorithm offers computational advantages over plain Extended Kalman Filter (EKF) based implementations. FastSLAM can associate data on a per-particle basis resulting in maintaining the posteriors over multiple data associations instead of the most likely one. Basic steps of the algorithm are given in Figure 5-3.

### 5.2.3 Obstacle Avoidance

Although obstacle avoidance can be treated as a separate topic, motion planning itself, includes collision avoidance by its nature. A map of the operation environment and a goal point is the primary inputs for a path planning algorithm. With these inputs, path planning algorithm computes a trajectory from the location of the robot to the goal point. This trajectory generally is the time-optimized path that considers the wall or other obstacles in the prior, static map. But today's robots do not move in static environments. They operate in unstructured roads, even off-road conditions; they need to plan a path in an environment where dynamic objects (like humans, animals, mobile devices etc.) are present. Therefore, a path planning system should operate in coordination with collision avoidance system that can detect obstacles not shown in static map and modify globally planned trajectory to avoid obstacles.

Collision detection changes robot's local path depending on the location of obstacles around. There are various different algorithms and approaches to obstacle avoidance issue. These algorithms interpret sensory information in different manners and generate different motion commands to avoid obstacles.

One of the most widely used path planning algorithms is potential fields method first proposed by Khatib [119]. In this method, goal destination generates potential field that attracts the robot while obstacles in the environment repulse the robot by potential fields they generate in opposing direction to themselves. Superposition of all the attractive and repulsive potential fields in the environment results in a path from current location to goal destination.

Methods which rely on potential fields generated by goal point and obstacles have some disadvantages. Most important of all, the robot which navigates according to a path generated by this kind of method may become stuck in s local minimum location Figure 5-4. This local minimum condition can be a result of symmetry in the map or concave shapes in the environment.

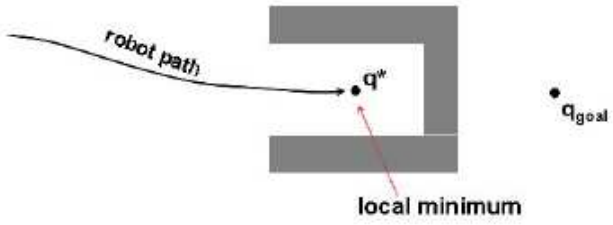


Figure 5-4: A local minimum example

Another method for obstacle avoidance is Vector Field Histogram (VFH) method, which is developed by Borenstein and Korem [120]. VFH has three main steps. In the first step, the algorithm builds a two dimensional histogram grid of the obstacles around the robot. In the second stage, the two dimensional histogram grid which is generated in first step is filtered according to the active window of the robot and a one dimensional polar histogram is generated. In the final step, previously generated one-dimensional polar histogram is used to generate steering angle and desired velocity commands. VFH is developed to overcome some advantages of potential fields methods described before. VFH does not accompany repulsive or attractive forces, therefore it does not cause the robot to be trapped in a local minimum.

Obstacle avoidance techniques in the literature are divided into two categories. First one is global techniques which require the complete map of the environment and is able to generate a path from current location to the goal position prior to movement. Most commonly used methods include potential field methods and cell decomposition [121]. If a global environment model is not available, then global methods are hard to implement. Also they are not suitable for collision avoidance when the robot moves fast.

The other obstacle avoidance techniques are generally referred as local and/or reactive techniques. Since they do not consider the global environment they generally cannot generate optimal collision avoidance trajectories. They use only a local circumference of the robot. But their computational complexity is relatively

low compared to global methods. This is the main advantage of local planning algorithms. One of their disadvantages is that they can be trapped in local minima present in the environment.

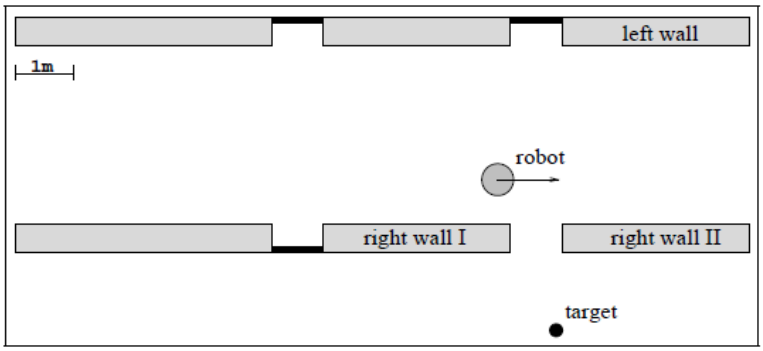


Figure 5-5: Problem example of disregarding of mechanical limitations [122]

Generally local planner algorithms do not consider the kinetic limitations of the robots. This results in excessive force requirement which a robot cannot generate in nature. An example to this situation is given in Figure 5-5. While moving relatively fast in a straight corridor, when the robot's local planner generates a path to the target located to the right of the robot, it generates a command set to turn the robot immediately in that direction. Since the vehicle dynamics will not allow this immediate turn, it will probably collide with "right wall II". A solution to this problem is given in Dynamic Window Approach which is presented in the next section [122].

.

- **Search space**: The search space of the possible velocities is reduced in three steps:
    - **Circular trajectories**: The dynamic window approach considers only circular trajectories (curvatures) uniquely determined by pairs (v, w) of translational and rotational velocities. This results in a two-dimensional velocity search space.
    - **Admissible velocities**: The restriction to admissible velocities ensures that only safe trajectories are considered. A pair (v, w) is considered admissible, if the robot is able to stop before it reaches the closest obstacle on the corresponding curvature.
    - **Dynamic window**: The dynamic window restricts the admissible velocities to those that can be reached within a short time interval given the limited accelerations of the robot.
- **Optimization**: The objective function

  $G(v,w) = \sigma(\alpha \cdot heading(v,w) + \beta \cdot dist(v,w) + \gamma \cdot vel(v,w))$

  is maximized. With respect to the current position and orientation of the robot this function trades of the following aspects:
    - **Target heading**: *heading* is a measure of progress towards the goal location. It is maximal if the robot moves directly towards the target.
    - **Clearance**: *dist* is the distance to the closest obstacle on the trajectory. The smaller the distance to an obstacle the higher is the robot's desire to move around it.
    - **Velocity**: *vel* is the forward velocity of the robot and supports fast movements.

The function $\sigma$ smoothes the weighted sum of the three components and results in more side-clearance from obstacles.

Figure 5-6: Parts of the dynamic window approach [122], [123]

#### 5.2.3.1 Dynamic Window Approach

Most important aspect of the dynamic window approach (DWA) is that, it incorporates the dynamics of the robotic asset. DWA generates velocity commands to control the robot regarding reachable velocity levels considering the dynamic constraints of the robot. Resultantly, DWA does not generate velocity commands that require infinite accelerations in an obstacle-free direction.

Objective function and steps of DWA algorithm are given in Figure 5-6. In the first stage it collects a set of possible velocity values. In the second stage the algorithm continues with an optimization. In this optimization phase, objective function is maximized and the corresponding velocity is selected from the set of the velocities generated in the first stage of the algorithm.

### 5.2.4 Software

Software framework and control architecture is represented in Section 3.12. As mentioned, Robot Operating System (ROS) is selected as the main software framework and basic control strategies for teleoperation of the UGV with a remote input device such as a steering wheel and pedal set are implemented. In this section, software developed and/or directly implemented for motion planning and collision avoidance during teleoperation missions with force feedback is described.

For navigation purposes, ROS navigation stack [124] is implemented in the software on the UGV computer. ROS navigation stack developed and maintained by Eltan Marder-Eppstein is a 2D navigation stack that takes in information from odometry system and sensor streams, and a navigation goal position and resultantly computes and outputs safe velocity commands for use with mobile robot base. An overview is presented in Figure 5-7.

Navigation stack provides some basic packages and nodes that are necessary for a robust path planning and navigation application. Besides these inherited packages,

additional platform-specific packages must be written and combined with navigation stack to make it work properly.

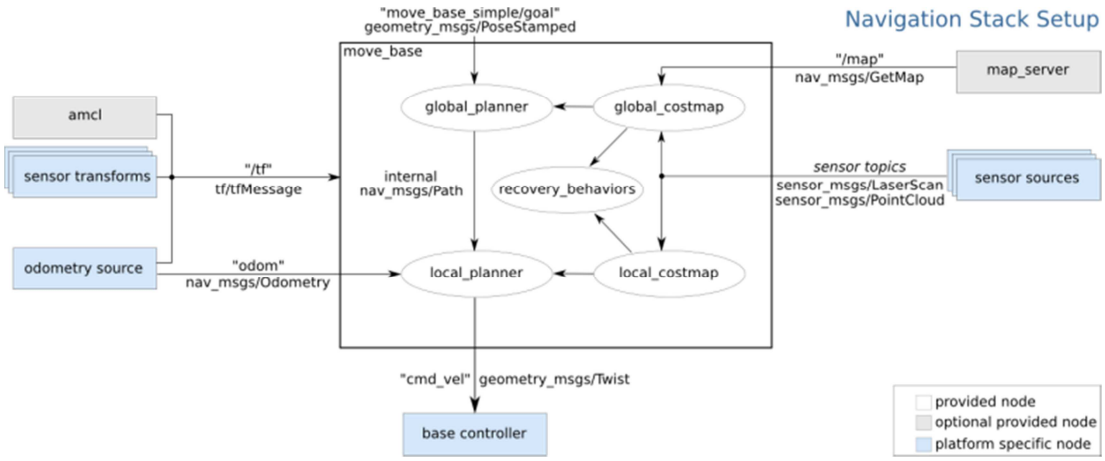


Figure 5-7: Overview of navigation stack

Navigation system has eight subsystems can be listed as,

- Mapping and Localization
- Odometry Source
- Sensor Sources
- Costmap
- Global Planner
- Local Planner
- Coordinate Frame Transformer
- Base Controller

#### 5.2.4.1 Mapping and Localization

Navigation stack of ROS can operate with a static map provided before run time. However, if a static map is not available, path planning works with only available obstacle information according to the location of the robot. Trajectories generated may, possibly, intersect an undetected obstacle. As the robot navigates, it adds the

information gathered to build its map and rebuilds the map to generate better plans that will now cause collisions. Since the robot is developed to move in unknown environments, a prior map is not available. Therefore, for mapping and localization, SLAM algorithm which is given in Section 5.2.2 is implemented. The algorithm is named as GMapping [125]. This approach uses a particle filter in which each particle carries an individual map of the environment. Accordingly, a key question is how to reduce the number of particles. In this algorithm, adaptive techniques to reduce the number of particles in a Rao-Blackwellized particle filter for learning grid maps is represented. This drastically decreases the uncertainty about the robot's pose in the prediction step of the filter.

#### 5.2.4.2 Odometry Source

Odometry information is basically gathered from proximity sensors located on the rear wheels of UGV. Transition durations from one notch to another on the encoder discs are used for estimating the vehicle speed and position. In this estimation, vehicle model represented in Section 5.2.1 is used. With this vehicle model, longitudinal and lateral velocities, and yaw rate of the vehicle with respect to body-fixed coordinate frame is obtained and converted to global coordinate frame to track the trajectory of the vehicle and supply position information to SLAM algorithm. Odometry information is published to “/odom” topic with the message type of “nav_msgs/Odometry”.

#### 5.2.4.3 Sensor Sources

The navigation system uses information from sensors to avoid obstacles in the world, and supply this information to SLAM algorithm to develop a map of the environment. This information needs to be supplied to develop costmaps as either “sensor_msgs/LaserScan” or “sensor_msgs/PointCloud” message type. The UGV has SICK LMS291 LIDAR system at front and the back of the chassis. These sensors are used as the primary sensor sources and the information from these laser range finders are published to “/scan”.

#### 5.2.4.4 Costmap

In the navigation system, local and global planners operate on a two dimensional planner Costmap. For the sake of simplicity and computational complexity, UGV is assumed to move on a planner surface. UGV can only overtake obstacles by moving around them, not by jumping over or stepping on. The costmap is not static. It starts with available sensory information on startup and updates and rebuilds itself whenever new information from sensor network is available. Costmap consists of grids; each has a cost depending on its occupancy. If an obstacle coincides with a grid on costmap, cost of according grid increases to a critical state that any part of robot footprint is not allowed to pass over. An inflation radius is defined in costmap configuration and grids with critical costs, meaning they include an obstacle, are inflated with exponentially decaying cost information in outwards direction from that grid.

#### 5.2.4.5 Global Planner

Global planner is responsible of generation high level plans from the most recent location of the robot to a goal position. Global planner makes use of costmap generated based on sensor readings and generates an optimum path, disregarding footprint of the robot due to efficiency and computational complexity constraints. The robot is assumed to have a circular shape to simplify the search algorithm. Since the UGV has Ackermann type steering system, ROS native global planner is replaced with Search-Based Planning Library (SBPL) Lattice Planner. It relies on graph search technique. ARA*, anytime version of A* search algorithm [126] is used to plan the optimum path. For use with SBPL global planner, motion primitives of the UGV is generated. There are mainly three motion primitives associated with the UGV: going straight forward, taking a curve to right or left with a minimum radius of curvature which is equal to turning radius of vehicle. Backwards motion is neglected to exclude it from the autonomous planning algorithm.

#### 5.2.4.6 Local Planner

Global path generated by the global planner is an input to local planner. Local planner generates local, short trajectories to avoid obstacles while moving as close as possible to the trajectory generated by the global planner. Local planner, unlike the global planner, considers the dynamic and kinematic limitations of the UGV and generates the necessary velocity commands for collision avoidance and path following. The technique used by the local planner is described in section 5.2.3.1, Dynamic Window Approach (DWA) [122]. DWA tries to minimize a cost function which includes costs of velocity, distance to obstacles and distance to global path. By adjusting the weighting factors of these components, the path following behavior of the UGV can be altered.

#### 5.2.4.7 Coordinate Frame Transformer

There are more than one coordinate frames in the navigation system. The global coordinate frame is the "/map" frame. This frame is world-fix which the trajectory of the robot is calculated against. Another frame "/odom" is the frame which odometry information is referenced on. In the frame tree, "/base_link" frame is present below the "/odom" frame. This frame is located at the center of gravity of the UGV. It is named as body-fixed coordinate frame also. The last coordinate frame is "/base_laser" whose origin is located at the center of the laser range finder located at the front of the vehicle. Coordinate frame transformer, transforms the laser measurements from "/base_laser" to "/base_link" frame and the resultant obstacle map is constructed regarding to this frame. Transformations are published to "/tf" topic with the message type of "tf/tfMessage".

#### 5.2.4.8 Base Controller

Local planner of the navigation system publishes the velocity commands which are computed according to local costmap to "/cmd_vel" topic. Message type for this publication is "geometry_msgs/Twist". It contains three velocity values which are $v_x$, $v_y$ and $v_\theta$. All these three velocity values are referenced with respect to body-

fixed coordinate frame. Base controller subscribes to "/cmd_vel" topic and gathers these information.

Since the UGV is teleoperated, this velocity commands are not fed to motor drivers. Instead they are used to guide the teleoperator for avoiding the obstacles and reducing his workload by this way. Next section is dedicated to force feedback implementation to the software.

### 5.3 Force Feedback

During teleoperation, due to limited field of view and insufficient situational awareness of the environment the UGV is located in, avoiding obstacles and navigating the vehicle is a challenging task. As proposed in earlier chapters, this study aims to reduce workload and increase task performance (i.e., taking the shortest path, navigating faster, feeling comfortable, hitting less obstacles etc.) of the operator by supplying force feedback aids for collision avoidance.

To provide force feedback to the operator, a steering wheel with force feedback capability is integrated into the system (Figure 3-32).

#### 5.3.1 Method

Command velocities that are produced by the local planner are used to compute necessary steering angle and the corresponding steering wheel angle. Then this steering wheel angle command is applied to the input device via force feedback, helping the operator to position the steering wheel to the computed angle to avoid upcoming obstacles. Operator can overcome this generated force and steer the vehicle to any other position he desires. However, the vehicle can steer itself to the computed direction if no counter force applied to the steering wheel.

Necessary steering angle is computed by the means of following equations. First, the desired radius of curvature by the local planner is found using two velocity values, $v_x$ and $v_y$. Using Eq. (5.3) and (5.4)

$$R = \frac{\sqrt{v_x^2 + v_y^2}}{v_\theta} \tag{5.15}$$

$$\delta_f = tan^{-1}\left(\frac{l_f + l_r}{R \cdot \cos(\beta)}\right) \tag{5.16}$$

This calculated steering angle is mapped to the steering wheel. There is a PID controller implemented to steering wheel force feedback system. It tries to position the wheel to the steering wheel angle corresponding to calculated steering angle.

### 5.3.2 Software

To implement the method described in the previous section, a ROS node has been written with the name of “force_feedback”. This node runs on UGV computer. It is subscribed to “/cmd_vel” topic. This node gets the velocity information from this topic. It uses Eq. (5.15) and (5.16) to calculate the necessary steering angle.

Once the steering angle needed to avoid upcoming obstacles, it is published to a topic named “/steering_angle”. Another node, written to control the steering wheel runs on remote control computer where the steering wheel is connected via USB. This node, “wheel_ff” namely, listens for steering angle commands from “force_feedback” node. If a command is received, this node loops the generic PID controller and generates necessary force to align the steering wheel with the desired angle.

## 5.4 Experiment

After successful development and implementation of whole control system, it is needed to be tested in real environments and with human participants. With this experiment, the developed UGV is validated in a force feedback application and effects of the force feedback enhancement on the teleoperation performance are observed. For the experiment, a set of participants is selected, a test scenario is constituted and the tests are performed. Results are logged and evaluated after on and they are described in Results section.

### 5.4.1 Method

For the experiment, a set of participants is selected. A teleoperation setup is prepared without line of sight of UGV and dependent variables are measured via surveys handed out before and after experiment.

#### 5.4.1.1 Participants

There are ten participants selected for the experiments. Participants are either research assistances in Mechanical Engineering Department of METU, or graduate students from the same university. Participants are males and their ages range from 23 to 28 with average value of 25.8 (SD = 1.33). 8 of 10 participants reported that they drive a car daily, 1 monthly and 1 never. Two participants mentioned that they are “experts”, 6 as “excellent” with computer usage while remaining mentioned as being “good”. 8 participants reported playing video games, especially racing or driving ones.

#### 5.4.1.2 Setup

Test base is selected as parking lot of Mechanical Engineering G-Building (Figure 5-8). This area is structured and the field is wide and empty enough for this experiment.


Figure 5-8: Test base located in department parking lot (39.889157, 32.779839)

In the test area, four packages each sized as 34 x 38 x 65 cm positioned with 1.5 meters sequentially. They had 1.5 meters in between also (Figure 5-9).

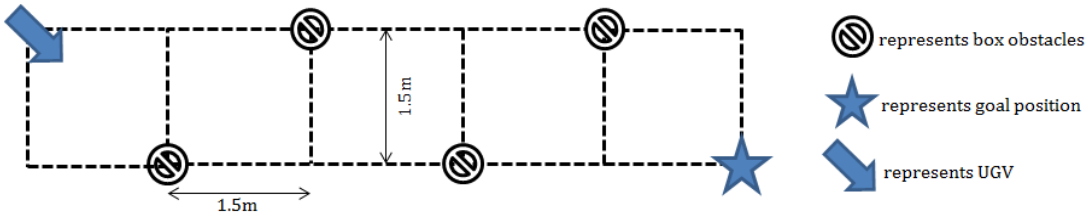


Figure 5-9: Experiment setup

A notebook computer with 15.6 inches screen size is used as the command computer. The steering wheel and the brake and throttle pedals are connected to this computer via USB port. Participants sit on a chair with control interfaces in front of them. They do not have a direct line of sight with the UGV. Video stream from the camera in front of the UGV is projected onto the notebook display in full screen mode. There are no other interface elements beside the camera stream. Only a time stamp is displayed in lower right corner of the screen (Figure 5-10).


Figure 5-10: Camera stream from the UGV

#### 5.4.1.3 Procedure

Before the experiment, participants are allowed to drive the UGV in the test route with or without line of sight to lessen learning effects. After this learning phase, demographic survey (see Appendix B) is handed out to the participant. Once the demographic survey is collected, test phase begins. Participant sits on the test bed in the building with no line of sight with the UGV. The UGV is positioned according to the experiment setup given in Figure 5-9. It is told to the user that he is required to navigate the UGV from starting point to the goal position without colliding with obstacles and wasting excessive time. After the confirmation of the participant, teleoperation mode without force feedback support is started and the user is informed that he can start to mission. He guides the vehicle using the full screen video stream on the remote computer screen.

Figure 5-11: Test setup prepared

When the goal point is reached, task completion time and number of objects collided are recorded. After completion of this first phase, second phase with force feedback support is started. A goal position is selected on the global costmap from remote computer. UGV plans a global path to the goal and local planner starts to give feedback to the steering wheel. Same path is taken and again same variables are

measured and recorded as in previous phase. After the end of the experiment, the participant is given a sheet of survey which is the NASA Task Load Index survey (see Appendix C) to measure the workload. This test procedure is applied to each participant and the resulting dependent variables are recorded. As dependent variables, task errors (number of boxes that collided), operator efficiency (time to complete task) and perceived workload (according to NASA-TLX scores) are selected.

### 5.4.2 Results

Once the tests are completed, results are collected and statistical data is generated from these results. Results section is divided into three as Task Completion Time, Number of Task Errors and Perceived Workload.

#### 5.4.2.1 Task Completion Time

The primary measure is the task completion time. Time is measured from the start of the movement of the UGV up to reaching to the goal point. If a time the test takes is longer than 60 seconds, the test is interrupted and the value is taken as 60. However none of the participants completed the mission beyond 40 seconds.

Without the proposed force feedback support, it took from 18.00 to 35.28 seconds to participants to complete the course. The mean task completion time was 22.41 seconds with a standard deviation of 4.94. With the presence of force feedback aid on the steering wheel, time to complete the task is reduced to a mean value of 19.72 seconds and a standard deviation of 4.88. The minimum task time with force feedback support was 15.2 seconds while the maximum was 34.12 (Table 5-1).

Table 5-1: Results of Task Completion Time

| **Task Type** | **Mean (sec)** | **St. Dev.** | **Minimum** | **Maximum** |
|---|---|---|---|---|
| Without FF | 22.41 | 4.94 | 18.00 | 35.28 |
| With FF | 19.72 | 4.88 | 15.20 | 34.12 |

#### 5.4.2.2 Number of Task Errors

The second measure of the operator performance is the number of task errors. Task error is defined as colliding with an obstacle on the course in this experiment. In the task with no force feedback aid, the operator only relies on the camera stream from the UGV and tries to navigate without colliding. With force feedback aid, the steering wheel applies a force to the operator in the direction resulted from the obstacle avoidance algorithms implemented. However the operator always can overcome the force and navigate the vehicle in totally different path from the UGV suggests.

Without the force feedback support, the participants collided with total of 7 obstacles on the path. Mean value is 0.7 with a standard deviation of 0.61. On the other side, with force feedback, only 1 collision is occurred which results in a mean of 0.1 and standard deviation of 0.29 (Table 5-2).

Table 5-2: Results of Number of Task Errors

| **Task Type** | **Mean (#)** | **St. Dev.** | **Total** |
|---|---|---|---|
| Without FF | 0.7 | 0.61 | 7 |
| With FF | 0.1 | 0.29 | 1 |

#### 5.4.2.3 Perceived Workload

Perceived workload of the operators is the final dependent variable to be measured. To measure the workload properly, Task Load Index of NASA (Appendix C) is used in both types of task (i.e. with and without force feedback).

It is seen that mental demand is more in the case of no force feedback. However physical demand is nearly same probably because of the force applied to the wheel in force feedback case. Participants feel more frustrated when there is no force feedback aid. Time pressure is less with force feedback support and the participants feel that

they are more successful with the force feedback support in accordance with the previous results in task completion time and task errors (Table 5-3).

Table 5-3: Results of NASA Task Load Index

| **Task Type** | **Mental Demand** | **Physical Demand** | **Temporal Demand** | **Level of Effort** | **Level of Frustration** | **Perform.** |
|---|---|---|---|---|---|---|
| W/O FF | 7.4 | 5.4 | 6.2 | 7.3 | 7.4 | 5.1 |
| W/ FF | 5.2 | 5.5 | 3.9 | 3.7 | 4.7 | 8.2 |

## 5.5 Conclusion

In this chapter, force feedback application the developed unmanned ground vehicle is described. Since the vehicle is developed with aim to use in autonomous algorithm research, a validation is done with this force feedback application.

Experiment results mentioned in previous sections show that the task completion time is shortened and task errors are lessened with introduction of the developed force feedback support to the operator in a teleoperation task. Results of perceived task load indicate that, the force feedback aid can be beneficial in terms of operator workload. Also these results validate that the developed UGV can be used in research studies especially with autonomous or semi-autonomous implementations.

# CHAPTER 5

# CONCLUSION

In this study, an ATV-sized unmanned ground vehicle is developed. Design and manufacturing of the main vehicle body and chassis are not included in the scope of this thesis but for the sake of completeness, body and chassis development is also included in a summarized manner. The two main parts of the study was to develop electrical and control systems of an unmanned ground vehicle, and to validate the developed systems with a force feedback aided obstacle avoidance implementation for better operator performance and decreased workload.

In the first part of the study, a detailed literature survey is conducted on the similar projects in the literature. Actuator types, power systems, processing hardware, environmental sensing equipment, state and localization sensors, communication means and software in the literature are all reviewed broadly to develop and implement advantageous solutions. Another review on the literature is carried out on the HRI topic. The operator workload during a teleoperation mission and proposed solutions are investigated deeply. It is seen that very limited research studies implemented the use of force feedback and haptic modality in addition to visual channels. It is thought that, addition of force feedback aid to a teleoperation task can improve the operator performance and decrease workload. Since the UGV is developed for further autonomy studies, it is decided to validate the vehicle with a force feedback teleoperation implementation.

The development phase of the vehicle consisted of steering, brake, throttle, power systems, processing hardware, environmental sensors, state sensing equipment, global localization hardware, communication link, and software and control

architecture. After completion of this phase, the UGV is made ready for basic teleoperation with steering wheel, brake and throttle pedals, and camera stream from the vehicle.

The second phase was the development of force feedback supported obstacle avoidance and navigation system to validate the developed UGV for use with further autonomy research. For this purpose, a multimodal user interface is developed implementing visual and haptic sensory channels of human. A motion planning system is developed and implemented in the vehicle control architecture. When the second phase is completed, tests are conducted with participants from the university. Dependent variables of the experiment were task completion time, number of task errors and perceived workload. Results of the experiment suggest that the developed UGV can be used for further studies in autonomy research. Besides, results indicate that addition of force feedback support during teleoperation can improve the operator performance and decrease workload.

At the end of the study, goals which are selected at the beginning of the study are accomplished. Electrical and control systems are developed in a generic manner that makes it applicable to other unmanned ground vehicle projects with any size. Also, experiments on the vehicle indicated that use of force feedback in teleoperation missions can be beneficial for task performance. Future studies may include more comprehensive experiments on the force feedback, or generally haptic, modalities introduced in teleoperation tasks. In addition to haptic modality, other modalities like auditive, olfactory, gustatory and vestibular modalities should be used and multimodal user interfaces should be developed in this manner. Also, level of autonomy can be increased to decrease the operator workload leading to the full autonomy which is the long term aim of the many research projects.

## REFERENCES


[1] Robotics Collaborative Technology Alliances, "Robotics collaborative technology alliances," in *Proceedings of Collaborative Technology Alliances Conference*, Washington DC, 2004.

[2] Karl Murphy and Steven Legowik, "GPS aided retrotraverse for unmanned ground vehicles," in *Proceedings of SPIE,* 1996, pp. 1-10.

[3] A. Trebi-Ollennu and J.M. Dolan, "An autonomous ground vehicle for distributed surveillance: cyberscout," *Robotics Institute*, p. 213, 1999.

[4] Team Ensco, "DARPA Grand Challenge Technical Paper," 2004.

[5] Team Spirit of Las Vegas, "DARPA Grand Challenge Technical Paper," 2004.

[6] K. D. Kuhnert and W. Seemann, "Design and realization of the highly modular and robust autonomous mobile outdoor robot Amor," *Robotics and Applications and Telematics,* 2007.

[7] DM Carroll, K Mikell, and Thomas Denewiler, "Unmanned ground vehicles for integrated force protection," Space and Naval Warfare Systems Center, San Diego, CA, 2004.

[8] G Bayar, İ Konukseven, B Koku, T Balkan, and A Erdener, "ATV Tabanlı İnsansız Kara Aracı Geliştirilmesi," *Makina Tasarım ve İmalat Dergisi*, vol. 8, no. 2, pp. 54-66, November 2007.

[9] A. Y. Phillip and Team MonsterMoto, "DARPA Grand Challenge Technical Paper," 2005.

[10] Overboat Team, "DARPA Grand Challenge Technical Paper," 2005.

[11] Cajunboat Team, "DARPA Grand Challenge Technical Paper," 2005.

[12] Chad Karl Tobler, "Development of an Autonomous Navigation Technology Test Vehicle," University of Florida, Master Thesis 2004.

[13] Lars B. Cremean, Tully B. Foote, Jeremy H. Gillula, and George H. Hines, "Alice: An information-rich autonomous vehicle for high-speed desert navigation," *Journal of Field Robotics*, vol. 23, no. 9, pp. 777-810, 2006.

[14] T Schönberg and M Ojala, "Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation," *International Journal of Systems Science*, vol. 27, no. 8, 1996.

[15] D. Wier, "Sequences of failure in complex socio-technical systems: Some implications of decision and control," *Kybernetes*, vol. 33, pp. 522-537, 2004.

[16] A. Klein, R.Watson, and C. Kitts C. Bulich, "Characterization of delay induced piloting instability for the Triton undersea robot," in *Proceedings of IEEE Aerospace Conf*, 2004, p. 423.

[17] Casper and R. R. Murphy, "Human–robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," *IEEE Trans. Syst. Man, Cybern. B: Cybern*, vol. 33, no. 3, pp. 367-385, 2003.

[18] C. Thorpe, and B. Glass T. Fong, "PdaDriver: A handheld system for remote driving," in *Proceedings of IEEE Int. Conf. Adv. Robot*, 2003.

[19] A. B. Koku, and S. Zein-Sabatto A. Sekmen, "Human robot interaction," in *Proceedings of IEEE Int. Conf. Syst., Man, Cybern.*, 2003, pp. 3937-3942.

[20] G. Kamsickas, "Future combat systems (FCS) concept and technology development (CTD) phase—Unmanned combat demonstration—Final report,"

Boeing Company, Seattle, WA, Tech. Rep. 2003.

[21] T. Fong and C.Thorpe, "Vehicle teleoperation interfaces," *Autonomous Robots*, vol. 11, pp. 9-18, 2001.

[22] H. K. Keskinpala and J. A. Adams, "Objective data analysis for a PDA-based human–robot interface," in *Proceedings of IEEE Int. Conf. Syst., Man.,Cybern.*, 2004, pp. 2809-2814.

[23] A. A. Malcolm and J. S. G. Lim. Teleoperation control for a heavy-duty tracked vehicle. Last accessed in March, 2012 [Online]. http://www.simtech.astar.edu.sg/Research/TechnicalReports/TR0369.pdf

[24] M. J. Barnes, J. Y. C. Chen, K. A. Cosenzo, and D. K.Mitchell, "Human robot teams as soldier augmentation in future battlefields: An overview," in *Proceedings of Int. Conf. Hum.–Comput. Interact.*, 2005.

[25] S. P. Schipani. An evaluation of operator workload during partially-autonomous vehicle operations. Last accessed in April, 2012 [Online]. http://www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Metrics/PerMIS_2003/Proceedings/Schipani.pdf

[26] M. J. Barnes, R. Parasuraman, and K. A. Cosenzo, "Adaptive automation for military robotic systems," *Uninhabited Military Vehicles: Human Factor Issues in Augmenting the Force*, pp. 420-440, 2006.

[27] Matthew S. Prewett, Ryan C. Johnson, Kristin N. Saboe, Linda R. Elliott, and Michael D. Coovert, "Managing workload in human–robot interaction: A review of empirical studies," *Computers in Human Behavior*, vol. 26, pp. 840-856, 2010.

[28] A. Steinfeld et al., "Common metrics for human-robot interaction," in *Proceedings of ACM Conf. Human-Robot Interact.*, 2006, pp. 33-40.

[29] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *IEEE Trans. Ind. Electron*, vol. 54, no. 4, pp. 699-704, 2003.

[30] D. D.Woods, J.Tittle, M. Feil, and A.Roesler, "Envisioning human-robot coordination in future operations," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev*, vol. 34, no. 2, pp. 210-218, 2004.

[31] J. S. Tittle, A. Roesler, and D. D. Woods, "The remote perception problem," in *Proceedings of Hum. Factors Ergonom. Soc. 46th Annu. Meet.*, 2002, pp. 260-264.

[32] A. K. Kanduri, G. Thomas, N. Cabrol, E. Grin, and R. C. Anderson, "The (in)accuracy of novice rover operators' perception of obstacle height from monoscopic images," *Trans. Syst. Man, Cybern. A, Syst.,Humans*, vol. 35, no. 4, pp. 505-512, 2005.

[33] R. P. Darken, K. Kempster, and B. Peterson, "Effect of streaming video quality of service on spatial comprehension in a reconnaissance task," in *Proceedings of 23rd Army Sci. Conf.*, 2001, pp. 391-400.

[34] R. R. Murphy, "Human–robot interaction in rescue robotics," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev*, vol. 34, no. 2, pp. 138-153, 2004.

[35] J. Carlson and R. R. Murphy. How UGVs physically fail in the field. Last accessed in June, 2012 [Online]. http://crasar.csee.usf.edu/ research/Publications/CRASAR-TR2004–16.pdf

[36] C. Wickens, "Multiple resources and performance prediction," *Theoretical Issues in Ergonomics Science*, vol. 3, no. 2, pp. 159-177, 2002.

[37] C. Wickens, "Multiple resources and mental workload," *Human Factors*, vol. 50, no. 3, pp. 449-454, 2008.

[38] G. Calhoun, M. Draper, J. Nelson, A. Lefebvre, and H. Ruff, "Simulation assessment of synthetic vision concepts for UAV operations," Air Force Research Laboratory, Wright-Patterson AFB, OH, Technical Report 2006.

[39] M. J. Massimino and T. B. Sheridan, "Teleoperator performance with varying force and visual feedback," *Human Factors*, vol. 36, no. 1, pp. 145-157, 1994.

[40] B. Watson, N. Walker, P. Woytiuk, and W. Ribarsky, "Maintaining usability during 3D placement despite delay," in *Proceedings of IEEE Conference on Virtual Reality*, 2003, pp. 133-140.

[41] B. Watson, N. Walker, W. Ribarsky, and V. Spaulding, "Effects of variation in system responsiveness on user performance in virtual environments," *Human Factors*, vol. 40, no. 3, pp. 403-414, 1998.

[42] P. Richard et al., "Effect of frame rate and force feedback on virtual object manipulation," *Presence*, vol. 5, no. 1, pp. 95-108, 1996.

[43] J. C. Lane et al., "Effects of time delay on telerobotic control of neutral buoyancy vehicles," in *Proceedings of IEEE international conference on robotics and automation*, Washington, DC, 2002, pp. 2874-2879.

[44] S. R. Ellis, K. Mania, B. D. Adelstein, and M. Hill, "Generalizeability of latency detection in a variety of virtual environments," in *Proceedings of Human factors and ergonomics society's 48th annual meeting*, New Orleans, LA, 2004, p. 2632.

[45] M. Draper, G. Calhoun, and J Nelson, "Evaluation of synthetic vision overlay concepts for UAV sensor operations: Landmark cues and picture-in-picture," Warfighter Interface Division, Wright-Patterson AFB, OH, Technical Report 2006.

[46] S. L. Pazuchanics, "The effects of camera perspective and field of view on performance in teleoperated navigation," in *Proceedings of Human factors and*

*ergonomics society's 50th annual meeting*, San Francisco, CA, 2006, pp. 1528-1532.

[47] D. R. Scribner and J. W. Gombash, "The effect of stereoscopic and wide field of view conditions on teleoperator performance," Army Research, Aberdeen Proving Grounds, MD, 1998.

[48] M. Lewis, J. Wang, J. Manojilovich, S. Hughes, and X. Liu, "Experiments with attitude: Attitude displays for teleoperations.," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 2003, pp. 1345-1349.

[49] T. Heath-Pastore, "Improved operator awareness of teleoperated land vehicle attitude," Naval Command, Control and Ocean Surveillance Center, CA, Technical Report 1994.

[50] O. Olmos, C. D. Wickens, and A. Chudy, "Tactical displays for combat awareness: An examination of dimensionality and frame of reference concepts and the application of cognitive engineering," *The International Journal of Aviation Psychology*, vol. 10, no. 3, pp. 247-271, 2000.

[51] L. C. Thomas and C. D. Wickens, "Effects of display frames of reference on spatial judgments and change direction," Army Research Laboratory, Aberdeen Proving Grounds, MD, Technical Report 2000.

[52] J. V. Draper, S. Handel, C. C. Hood, and C. T. Kring, "Three experiments with stereoscopic television: when it works and why," *IEEE transactions on systems, man and cybernetics*, pp. 1047-1052, 1991.

[53] D. M. Lion, "Three dimensional manual tracking using a head-tracked stereoscopic display," Human Interface Technology Lab, WA, Technical Report 1993.

[54] S. H. Park and J. C. Woldstad, "Multiple two-dimensional displays as an alternative to three-dimensional displays in telerobotic tasks," *Human Factors*, vol. 42, no. 4, pp. 592-603, 2000.

[55] C. W., & Goodrich, M. A. Nielson, "Comparing the usefulness of video and map information in navigation tasks," in *Proceedings of Human Robot Interaction*, Salt Lake City, UT, 2006, pp. 95–101.

[56] J. Y. C. Chen and C. Joyner, "Concurrent performance of gunner's and robotics operator's tasks in a multi-tasking environment," *Military Psychology*, vol. 21, pp. 98-113, 2009.

[57] B. Hardin and M. Goodrich, "On using mixed-initiative control: A perspective for managing large-scale robot teams," in *Proceedings of ACM/IEEE international conference on human–robot interaction*, La Jolla, CA, 2009, pp. 165-172.

[58] B. G. Witmer and P. B. Kline, "Judging perceived and traversed distance in virtual environments," *Presence*, vol. 7, no. 2, pp. 144-167, 1998.

[59] M. Yeh and C. D. Wickens, "Display signaling in augmented reality: Effects of cue reliability and image realism on attention allocation and trust calibration," *Human Factors*, vol. 43, no. 3, pp. 355-365, 2001.

[60] A. Fisher, P. L. McDermott, and S. Fagan, "Bandwidth allocation in a military teleoperation task," in *Proceedings of Human factors and ergonomics society's 53rd annual meeting*, La Jolla, CA, 2009, pp. 287–288.

[61] R. P. Darken and H. Cervik, "Map usage in virtual environments: Orientation Issues," in *Proceedings of IEEE Conference on Virtual Reality*, 1999, pp. 133-140.

[62] D. J. Folds and J. Gerth, "Auditory monitoring of up to eight simultaneous sources," in *Proceedings of Human factors and ergonomics society's 38th*

*annual meeting*, Santa Monica, CA, 1994, pp. 505-509.

[63] R. Parasuraman, S. M. Gaslter, P. Squire, H. Furukawa, and C. Miller, "Flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with RoboFlag," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 4, pp. 481-493, 2005.

[64] C. C. Smyth, J. W. Gombash, and P. M. Burcham, "Indirect vision driving with fixed flat panel displays for near-unity, wide, and extended fields of camera view," Army Research Laboratory, Aberdeen Proving Grounds, MD, Technical Report 2001.

[65] J. L. Burke et al., "Comparing the effects of visual-auditory and visual-tactile feedback on user performance. A meta-analysis," in *Proceedings of International Conference on Multimodal Interfaces*, 2006, pp. 108-117.

[66] J. Y. C. Chen, E. Haas, and M. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1231-1245, 2007.

[67] M. D. Coovert, A. Walvoord, L. Elliott, and E. Redden, "A tool for the accumulation and evaluation of multimodal research," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 5, pp. 1884-1906, 2008.

[68] M. S. Prewett et al., "The benefits of multimodal information: A meta-analysis comparing visual and visual-tactile feedback," in *Proceedings of International Conference on Multimodal Interfaces*, 2006, pp. 333-338.

[69] S. R. Dixon and C. D. Wickens, "Control of multiple-UAVs: A workload analysis," in *Proceedings of International symposium on aviation psychology*, Dayton, OH, 2003, pp. 1-5.

[70] C. D. Wickens, S. R. Dixon, and D. Chang, "Using interference models to predict performance in a multiple-task UAV environment- 2 UAVs," University of Illinois, Urbana-Champaign, Technical Report 2003.

[71] S. M. Galster, B. A. Knott, and R. D. Brown, "Managing multiple UAVs: Are we asking the right questions?," in *Proceedings of Human factors and ergonomics society's 50th annual meeting*, San Francisco, CA, 2006, pp. 545-549.

[72] K. A. Cosenzo, R. Parasuraman, A. Novak, and M. Barnes, "Implementation of automation for control of robotic systems," Army Research Laboratory, Aberdeen Proving Grounds, MD, Technical 2006.

[73] M. Draper, G. Calhoun, H. Ruff, D. Williamson, and T. Barry, "Manual versus speech input for unmanned aerial vehicle control stations," in *Proceedings of Human factors and ergonomics society's 47th annual meeting*, Denver, CO, 2003, pp. 387-391.

[74] K. L. Mosier, N. Sethi, S. McCauley, L. Khoo, and J. M. Orasanu, "What you don't know can hurt you: Factors impacting diagnosis in the automated cockpit," *Human Factors*, vol. 49, no. 2, pp. 300-310, 2007.

[75] K. Hendy, J. Lao, and P. Milgram, "Combining time and intensity effects in assessing operator information-processing load," *Human Factors*, vol. 30, pp. 30-47, 1997.

[76] J. J. Adams, "Multiple robot/single human interaction: Effects on perceived workload," *Behavior Information and Technology*, vol. 28, no. 2, pp. 183-198, 2009.

[77] R. A. Chadwick, "The impacts of multiple robots and display views: An urban search and rescue simulation," in *Proceedings of Human factors and ergonomics society's 49th annual meeting*, Orlando, FL, 2005, pp. 387-391.

[78] S. G. Hill and B. Bodt, "A field experiment of autonomous mobility: Operator workload for one or two robots," in *Proceedings of ACM/IEEE international conference on human–robot interaction*, Arlington, VA, 2007, pp. 169-176.

[79] J. W. Crandall and M. L. Cummings, "Developing performance metrics for the supervisory control of multiple robots," in *Proceedings of ACM/IEEE international conference on human–robot interaction*, Arlington, VA., 2007, pp. 33-40.

[80] P. Lif, H. Jander, and J. Borgwall, "Tactical evaluation of unmanned ground vehicle during a MOUT exercise," in *Proceedings of 7th international conference of engineering psychology and cognitive ergonomics*, Beijing, China, 2007, pp. 731-740.

[81] P. Squire, G. Trafton, and R. Parasuraman, "Human control of multiple unmanned vehicles: effects of interface type on execution and task switching times," in *Proceedings of 1st ACM SIGCHI/SIGART conference on human–robot interaction*, Salt Lake City, UT, 2006, pp. 26-32.

[82] R. A. Chadwick, "Operating multiple semi-autonomous robots: Monitoring, responding, detecting," in *Proceedings of Human factors and ergonomics society's 50th annual meeting*, San Francisco, CA, 2006, pp. 329-333.

[83] B. R. Levinthal and C. D. Wickens, "Management of multiple UAVs with imperfect automation," in *Proceedings of Human factors and ergonomics society's 50th annual meeting*, San Francisco, CA, 2006, pp. 1941-1944.

[84] T. B. Sheridan and W. L. Verplank, "Human and Computer Control of Undersea Teleoperators," MIT Man-Machine Systems Laboratory, Cambridge, MA, Technical Report 1978.

[85] D. J. Bruemmer, R. L. Boring, D. A. Few, J. L. Marble, and M. C. Walton, ""I call shotgun!": An evaluation of mixed-initiative control for novice users of a search and rescue robot," *IEEE Transactions on Systems, Man, and*

*Cybernetics*, vol. 34, pp. 2847-2852, 2004.

[86] M. R. Endsley and D. B. Kaber, "Level of automation effects on performance, situation awareness and workload in a dynamic control task," *Ergonomics*, vol. 42, pp. 462-492, 1999.

[87] S. Hughes and M. Lewis, "Task-driven camera operations for robotic exploration," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 4, pp. 513-522, 2005.

[88] D. B. Kaber, E. Onal, and M. Endsley, "Design of automation for telerobots and the effect on performance, operator situation awareness and subjective workload," *Human Factors and Ergonomics in Manufacturing*, vol. 10, pp. 409-430, 2000.

[89] J. P. Luck, P. L. McDermott, L. Allender, and D. C. Russell, "An investigation of real world control of robotic assets under communication latency," in *Proceedings of Human Robot Interaction*, Salt Lake City, UT, 2006, pp. 202-209.

[90] J. Y. C Chen, "Concurrent performance of military tasks and robotics tasks: Effects of automation unreliability and individual differences," in *Proceedings of The fourth human robot interaction conference*, La Jolla, CA, 2009, pp. 181-188.

[91] S. R. Dixon and C. D. Wickens, "Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload," *Human Factors*, vol. 48, pp. 474-486, 2006.

[92] E. K. Muthard and C. D. Wickens, "Factors that mediate flight plan monitoring and errors in plan revision: Planning under automated and high workload conditions," in *Proceedings of 12th international symposium on aviation psychology*, Dayton, OH, 2003.

[93] E. Rovira, K. McGarry, and R. Parasuraman, "Effects of imperfect automation on decision making in a simulated command and control task," *Human Factors*, vol. 49, pp. 76-87, 2007.

[94] Truck And Bus Total Vehicle Steering Committee, "Turning Ability and Off Tracking--Motor Vehicles," SAE, J695, 2011.

[95] H. J. Arditty and H. C. Leèfovre, "Sagnac effect in fiber gyroscopes," *Optics Letters*, vol. 6, no. 8, pp. 401-403, 1981.

[96] M. Quigley, B. Gerkey, and et al., "ROS: an open-source Robot Operating System," in *Proceedings of Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)*, Kobe, 2009.

[97] ROS Wiki - Robots. Last accessed in September, 2012 [Online]. http://www.ros.org/wiki/Robots

[98] sicktoolbox_wrapper - ROS Wiki. Last accessed in June, 2012 [Online]. http://www.ros.org/wiki/sicktoolbox_wrapper

[99] Real Time DAQ Linux for Humusoft's I/O devices. Last accessed in August, 2012 [Online]. http://www.penguin.cz/~fojtik/hudaqlib/hudaqlib.html

[100] Isaac Asimov, "Runaround," *Astounding Science Fiction*, March 1942.

[101] M. A. Goodrich and Jr. D. R. Olsen, "Seven principles of efficient human robot interaction," in *Proceedings of IEEE Int. Conf. Syst., Man, Cybern.*, 2003, pp. 3943-3948.

[102] D. K. Schreckenghost, "Checklists for human-robot collaboration during space operations," in *Proceedings of 43rd Annual Meeting of the Human Factors and Ergonomics Society*, Santa Monica, CA, 1999, pp. 46-50.

[103] J. Scholtz and S. Bahrami, "Human-robot interaction: Development of an evaluation methodology for the bystander role of interaction," in *Proceedings*

*of IEEE Int. Conf. Systems, Man, Cybernetics*, 2003, pp. 3212–3217.

[104] D. D. Woods, J. Tittle, M. Feil, and A. Roesler, "Envisioning human-robot coordination in future operations," *IEEE Transactions on Systems, Man & cybernetics: Part C: Special Issue on Human-Robot Interaction*, 2004.

[105] J. S. Tittle, A. Roesler, and D.D. Woods, "The remote perception problem," in *Proceedings of Human Factors and Ergonomics Society 46th Annual Meeting*, Santa Monica, CA, 2002, pp. 260-264.

[106] D. Drascic, "An investigation of monoscopic and stereoscopic video for teleoperation," University of Toronto, Master Thesis 1991.

[107] A. Steinfeld et al., "Common metrics for human-robot interaction," in *Proceedings of Int. Conf. Human-Robot Interaction*, Salt Lake City, UT, 2006, pp. 33-40.

[108] L. Nigay and J. Coutaz, "A design space for multimodal systems: concurrent processing and data fusion," in *Proceedings of Human Factors in Computer Systems*, 1993, pp. 172-178.

[109] Stefan Silbernagel and Agamemnon Despopoulos, *Taschenatlas der Physiologie*. Berlin, Gemany: Thieme, 1979.

[110] L. Schomaker and et al., "A Taxonomy of Multimodal Interaction in the Human Information Processing System," Esprit Basic Research Action Miami, 1995.

[111] M.T. Maybury and W. Wahlster, *Readings in Intelligent User Interfaces*.: Morgan Kaufmann, 1998.

[112] Merriam-Webster Online Dictionary. Last accessed in August, 2012 [Online]. http://www.merriam-webster.com/dictionary/haptic

[113] Kevin J Connolly and Jaan Valsiner, *Handbook of Developmental Psychology*.: SAGE, 2003.

[114] H.Z. Tan, B. Eberman, M.A. Srinivasan, and B. Cheng, "Human factors for the design of force-reflecting haptic interfaces," in *Proceedings of ASME Dynamic Systems and Control Division*, 1994, pp. 353-359.

[115] Danwei Wang and Feng Qi, "Trajectory Planning for a Four-Wheel-Steering Vehicle," in *Proceedings of IEEE International Conference on Robotics & Automation*, Seoul, 2001, pp. 3320-3325.

[116] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*. USA: MIT Press, 2005.

[117] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," in *Proceedings of AAAI National Conference on Artificial Intelligence*, Edmonton, 2002.

[118] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[119] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1995.

[120] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transaction on Robotics and Automation*, vol. 7, no. 3, pp. 278-288, 1991.

[121] J. Latombe, *Robot Motion Planning*. Boston, MA, USA: Kluwer Academic Publishers, 1991.

[122] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, "The dynamic window

approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23-33, March 1997.

[123] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proceedings of IEEE Int. Conf. Robotics and Automation*, 1996.

[124] Eitan Marder-Eppstein. ROS Navigation Stack. Last accessed in August, 2012 [Online]. http://www.ros.org/wiki/navigation

[125] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34-46, 2006.

[126] K. Konolige, "A gradient method for realtime robot control," in *Proceedings of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2000.

[127] J. Y. C. Chen, P. J. Durlach, J. A. Sloan, and L. D. Bowers, "Human–robot interaction in the context of simulated route reconnaissance missions," *Military Psychology*, vol. 20, pp. 135-149, 2008.

[128] M. Reddy, "The effects of low frame rate on a measure of user performance in virtual environments," Dept of Computer Science, University of Edinburgh, Technical Report 1997.

[129] B. D. Adelstein, G. L. Thomas, and S. R. Ellis, "Head tracking latency in virtual environments: Psychophysics and a model," in *Proceedings of Human factors and ergonomics society 47th annual meeting*, 2003, pp. 2083-2087.

[130] R. S. Allison, J. E. Zacher, D. Wang, and J. Shu, "Effects of network delay on a collaborative motor task with telehaptic and televisual feedback," in *Proceedings of ACM SIGGRAPH international conference on virtual reality continuum and its applications in industry*, 2004, pp. 375-381.

[131] M. A. Shreik-Nainar, D. B. Kaber, and M.-Y. Chow, "Control gain adaptation

in virtual reality mediated human–telerobot interaction," *Human Factors and Ergonomics in Manufacturing*, vol. 15, no. 3, pp. 259-274, 2003.

[132] R. Parasuraman, S. Galster, and C. Miller, "Human control of multiple robots in the RoboFlag simulation environment," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 33, no. 4, pp. 3233-3237, 2003.

[133] C. C. Smyth, "Modeling indirect vision driving with fixed flat panel displays: Task performance and mental workload," Army Research Laboratory, Aberdeen Proving Grounds, MD, Technical Report 2002.

[134] W. Wang and P. Milgram, "Effects of viewpoint displacement on navigational performance in virtual environments," in *Proceedings of Human factors and ergonomics society 47th annual meeting*, Denver, CO, 2003, pp. 139-143.

[135] J. L. Drury, B. Keyes, and H. A. Yanco, "LASSOing HRI: Analyzing situation awareness in map-centric and video-centric interfaces," in *Proceedings of ACM/IEEE international conference on human–robot interaction*, Arlington, VA, 2007, pp. 279-286.

[136] S. A. Murray, "Human–machine interaction with multiple autonomous sensors," in *Proceedings of IFAC/IFIP/IFORS/IEA symposium on analysis design and evaluation of man–machine systems*, 1995.

[137] D. Drascic and J. Grodski, "Defense teleoperation and stereoscopic video," in *Proceedings of SPIE: Stereoscopic Displays and Applications IV*, 1993, pp. 1-12.

[138] B. P. Sellner, L. M. Hiatt, R. Simmons, and S. Singh, "Attaining situational awareness for sliding autonomy," in *Proceedings of ACM SIGCHI/SIGART conference on human–robot interaction*, Salt Lake City, UT, 2006.

[139] J. Wang, H. Wang, and M. Lewis, "Assessing cooperation in human control of heterogeneous robots," in *Proceedings of The third ACM/IEEE international*

*conference on human–robot interaction*, Amsterdam, Netherlands, 2008, pp. 9-15.

[140] H. Wang, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara, "How search and its subscale tasks in N robots," in *Proceedings of ACM/IEEE international conference on human–robot interaction*, La Jolla, CA, 2009, pp. 141-147.

[141] B. Trouvain and C. Wolf, "Design and evaluation of a multi-robot interface," in *Proceedings of The role of humans in intelligent and automated systems*, Warsaw, Poland, 2003, pp. 1-12.

[142] B., Schlick, C., & Mervert, M. Trouvain, "Comparison of a map vs. camera based user interface in a multi-robot navigation task.," in *Proceedings of IEEE interantional conference on systems, man and cybernetics*, Washington DC, USA, 2005, pp. 3224-3231.

[143] L. Wang, G. A. Jamieson, and J. G. Hollands, "Trust and reliance on an automated combat identification system," *Human Factors*, vol. 51, pp. 281-291, 2009.

[144] D. B., & Endsley, M. R. Kaber, "The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task," *Theoretical Issues in Ergonomics Science*, pp. 1-40, 2003.

[145] E. Krotkov, R. Simmons, F. Cozman, and S. Koenig, "Safeguarded teleoperation for lunar rovers: From human factors to field trials," in *Proceedings of 26th international conference on environmental systems*, Santa Monica, CA, 1996.

[146] P. Schermerhorn and M. Schultz, "Dynamic robot autonomy: Investigating the effects of robot decision-making in a human-robot team task," in *Proceedings of the 2009 international conference on multimodal interfaces*, Cambridge,

MA, 2009, pp. 63-70.

[147] J. Wang and M. Lewis, "Human control for cooperating robot teams," in *Proceedings of The second human–robot interaction conference*, Arlington, VA, 2007, pp. 9-16.

[148] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall, "Managing autonomy in robot teams: Observations from four experiments," in *Proceedings of the second human robot interaction conference*, Arlington, VA, 2007, pp. 25-32.

[149] J. Meyer, L. Feinshreiber, and Y. Parmet, "Levels of automation in a simulated failure detection task," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 33, no. 3, pp. 2101-2106, 2003.

[150] H. A. Ruff, S. Narayan, and M. H. Draper, "Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles," *Presence*, vol. 11, no. 4, pp. 335-351, 2002.

[151] C. D. Wickens, S. R. Dixon, J. Goh, and B. Hammer, "Pilot dependence on imperfect diagnostic automation in simulated UAV flights: An attentional visual scanning analysis," University of Illinois, Urbana-Champaign, Technical Report 2005.

[152] J. Y. C. Chen and J. E. Thropp, "Review of low frame rate effects on human performance," *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1063-1076, 2007.

[153] L. R. Elliott et al., "A meta-analysis of vibrotactile and visual information displays for improving task performance," Army Research Laboratory, Fort Benning, GA, Unpublished manuscript 2009.

[154] E. A. Locke and G. P. Latham, *A theory of goal setting and task performance*.

Englewood Cliffs, NJ, USA: Prentice-Hall, 1990.

[155] H. Wang et al., "Human teams for large scale multirobot control," in *Proceedings of IEEE international conference on systems, man, & cybernetics*, San Antonio, TX, 2009, pp. 1269-1274.

[156] H. A. Ruff, G. Calhoun, M. Draper, J. Fontejon, and J. Abbott, "Exploring automation issues in supervisory control of multiple UAVs," in *Proceedings of 5th conference of human performance, situation awareness, and automation technology*, 2004, pp. 218-222.

[157] S. Beer, *Decision and control: The meaning of operational research and management cybernetics*. Chichester: John Wiley, 1966.

[158] M. D. Coovert, K. Ramakrishna, and E. Salas, "Preferences for power in expert systems by novice users," *AI & Society*, pp. 59-61, 1989.

[159] A. M. Cassidy, "Mental models, trust, and reliance: Exploring the effect of human perceptions on automation use," Naval Post-Graduate School, Master's Thesis 2009.

[160] G. Klein, R. M. Pliske, B. Crandall, and D. Woods, "Features of problem detection," in *Proceedings of The human factors and ergonomics society's 43rd annual meeting*, Houston, TX, 1999, pp. 133-137.

[161] K. J. Vicente, *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*. Mahwah, NJ: LEA, 1999.

[162] S. R. Dixon, C. D. Wickens, and D. Chang, "Comparing quantitative model predictions to experimental data in multiple-UAV flight control.," in *Proceedings of Human Factors and Ergonomics Society 47th Annual Meeting*, Santa Monica, CA, 2003, pp. 104-108.

[163] D. R. Olsen and M. A. Goodrich, "Metrics for evaluating human-robot

interactions," in *Proceedings of NIST Performance Metrics for Intelligent Systems Workshop*, 2003.

# APPENDIX A

## SUMMARY TABLES OF STUDIES RELATED TO WORKLOAD

Table A-1: Summary of studies manipulating frame rate [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Calhoun et al. [38] | 7 update rates: 0.5–24 Hz | Efficiency, SA, usability, and workload on UAV targeting | – Higher update rates improved subjective performance ratings<br>– No difference on efficiency between FR conditions |
| Chen et al. [127] | Normal vs. degrading: from 25 to 5 frames per second (fps) | Errors, efficiency, usability, workload, and sickness on UAV and UGV navigation and targeting | – No significant differences between presence or lack of<br>– Usability decreased with presence of latency |
| Darken et al. [33] | 4 Update rates: 1.5–22 fps | Errors, SA, and usability during building navigation | – No significant differences found between FR video conditions; no significant learning effects |
| Fisher et al. [60] | Resolution-FR combination | Usability (FR/resolution combination preference) | – Combination of high resolution/low frame rate was used most often (5 combinations from high res/low FR to low res/high FR) |
| Lion [53] | 33 vs. 22 Hz | Errors on a tracking task using 3D interface | – Higher FR related to better performance; learning effects present |
| Massimino, Sheridan [39] | 3 fps vs. 5 fps vs. 30 fps | Efficiency in moving mechanical arm to target via camera view | – Increased FR significantly improved efficiency; the addition of force feedback improved efficiency for all FR conditions |
| Reddy [128] | A: 2.3 vs. 11.5 Hz<br>B: 6.7 vs. 14.2 Hz | Errors and efficiency in completing a VE navigation task | – Errors and efficiency decreased with lower FR |
| Richard et al. [42] | 6 Update rates: 1–25 fps | Efficiency in tracking and grasping 3-D moving virtual target | – Higher FR coupled with MS compensated for a lack of SS visual cues; learning effects were significant |
| Watson et al. [41] | 3 studies: 9 Hz vs. 13 Hz vs. 17 Hz | Efficiency, errors, RT, and usability on grasping of virtual object using HMD | – With lower FR, RT increased, usability decreased and efficiency was reduced; errors were not significantly effected |
| Watson et al. [40] | 35, 75, 115 ms | Errors, efficiency, and usability on virtual object placement (HMD) | – Efficiency decreased and errors and task difficulty increased as FR decreased |

Table A-2: Summary of studies manipulating latency [27].

| **Study** | **Manipulation** | **Criteria** | **Results** |
|---|---|---|---|
| Adelstein et al. [129] | Latency, Constant or random head motion rates | RT to stimuli in VE using HMD | – Only interactions were significant<br>– Changes in motion patterns resulted in a decrease in operators' discrimination abilities and latency detection |
| Allison et al. [130] | Latency delay between 2 workstations | Errors, efficiency | – Greater system latency delays reduced efficiency, increased error rates and increase the time spent making errors |
| Chen et al. [127] | Normal vs. 250 ms delay | Errors, efficiency, usability, workload, and sickness on UAV and UGV navigation and targeting | – No significant differences between FR conditions for UAV;<br>– For UGVs, performance (hit rates) decreased with reduced FR |
| Ellis et al. [44] | Latency detection | Errors and efficiency in latency detection of VE with a HMD | - Complexity of environment failed to effect operator errors;<br>learning effects reported |
| Lane et al. [43] | Time delay between input and robot action | Efficiency in tracking and grabbing using UGV simulator | – Increased time delays led to a decrease in efficiency |
| Luck et al. [89] | Study A and B: Latency rates, variable and fixed latency lengths | Errors, efficiency, and usability in navigation on UGV simulator | – Increased latency/time delay let to a reduction in efficiency and more errors; efficiency improved when time delay was fixed as opposed to variable |
| Shreik-Nainar et al. [131] | Constant or random time delay | Errors and efficiency in navigation of VE with a HMD | – When time delay was constant, as opposed to variable, errors increased and efficiency decreased |
| Watson et al. [40] | Image latency, system responsiveness | Errors and efficiency in VE navigation using HMD | – Significant learning effects for impact of system latency |

Table A-3: Summary of studies manipulating field of view (FOV) [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Draper et al. [45] | Narrow vs. Wide | Efficiency, errors, and usability on UGV search task | – Completion times were faster with a wider FOV; efficiency is incrementally improved when both wide FOV and warning are present |
| Parasuraman et al. [132] | Visual range of camera | Efficiency and workload in virtual UGV navigation | – FOV showed no effects on criteria |
| Parasuraman et al. [63] | FOV at 3 levels (Narrow–Wide) | Efficiency, workload, and SA in UGV navigation of VE | – Workload increased as FOV decreased; no significant difference was present for efficiency |
| Pazuchanics [46] | Narrow vs. Wide | Efficiency, errors, and usability in UGV navigation | – Widening FOV resulted in improved performance compared to narrower FOV |
| Reddy [128] | 2 Studies: 8 levels of FOV (0.25°–32°) | Efficiency and errors on navigation task in VE | – Errors and efficiency were reduced with wider FOV |
| Scribner and Gombash [47] | Narrow vs. Wide | Errors, efficiency, stress and motion sickness in UAV navigation | – Motion sickness was reported more frequently in wide FOV condition; no interaction was present between FOV and depth cues |
| Smyth et al. [64] | Direct vs. 3 indirect view types (unity, wide, extended) | Errors, efficiency, workload, stress, and sickness on UGV navigation | – Wider FOV was desired for navigation but the FOV closest to typical vision was preferred for steering |
| Smyth [133] | Indirect vs. natural vs. unity | Errors, efficiency, workload, stress and sickness on UGV navigation | – Indirect FOV resulted in decreased driving speed and more errors compared to the baseline natural vision condition |
| Wang and Milgram [134] | 6 Comparisons of FOV | Errors and SA in navigation of UGV | – SA increased as FOV extended outward from robot; the moderate – FOV condition provided the best local SA and error rate |

Table A-4: Summary of studies manipulating camera perspective [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Darken and Cervik [61] | Map direction orientation | Errors and efficiency in UGV navigation task using camera/map | – Forward-up map alignment was best for targeted searches but north-up alignment was best for naïve and primed searches |
| Draper et al. [45] | Camera view vs. picture-in-picture | Efficiency, errors, and usability on UGV search task | – Usability was reduced when camera perspective is placed within the virtual environment display (picture-in-picture) |
| Drury et al. [135] | Map-based vs. video-based display | Errors, efficiency, SA, and usability for UGV search and navigation | – Video-based displays provided better performance indices, but map-based displays yielded better location and status awareness |
| Heath-Pastore [49] | Gravity-based vs. vehicle-based | Errors in navigation of UGV simulator | – Operators reported greater confidence and SA for gravity-referenced view; gravity-based perspective also yielded fewer errors |
| Hughes and Lewis [87] | Camera alignment and # of cameras | Errors and usability in UGV navigation and target identification | – Operator controlled cameras best for usability |
| Lewis, et al. [48] | Gravity-based vs. vehicle-based | Errors, efficiency, and usability in navigation of UGV | – Efficiency and usability were significantly better for gravity-fixed display |
| Murray [136] | Fixed vs. mobile vehicle-based view | Efficiency on target detection using camera views | – Efficiency was reduced with mobile camera views versus fixed-position cameras |
| Nielson and Goodrich [55] | Video-only, map-only, or video-map | Errors and efficiency in UAV navigation | – Video-only displays yielded slower completion times than the other two conditions, particularly when display was 2-D |
| Olmos et al. [50] | Exocentric vs. split-screen display | Error, Efficiency, and RT for navigation of VR terrain | – Split-screen, when displays were made visually consistent, yielded stronger performance indices than 2D and 3D exocentric displays |
| Thomas and Wickens [51] | Third person view vs. first person | Errors, RT, and usability for navigation of UGV simulator | – Third person view yielded faster RT, fewer errors and operators reported higher levels of confidence (usability) compared to the first person view |

Table A-5: Summary of studies manipulating depth cues [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Drascic and Grodski [137] | SS vs. MS | Navigation errors with robot arm | – SS display significantly reduced errors compared to MS display |
| Draper et al. [52] | 3 Studies: SS vs. MS | Errors and efficiency during placement task using robot arm | – SS displays provided better performance indices than MS displays in difficult conditions only |
| Lion [53] | SS vs. MS | Production and errors on 3D tracking task | – SS display was significantly related to enhanced performance and a reduction in errors |
| Nielson et al. [55] | 2-D vs. 3-D cues across display types | Errors and efficiency in UAV navigation | – Map-only display had slower completion times than map-video (2D) and video-only (3D); learning effects were detected |
| Olmos et al. [50] | 2-D vs. exocentric 3-D and split-screen 3-D displays | Error, efficiency, & RT for navigation of VR terrain | – 2D display was detrimental to vertical maneuver performance, 3D display showed greatest deficits during lateral maneuvers |
| Park and Woldstad [54] | 2-D vs. 3-D MS vs. 3-D SS | Errors, efficiency, and workload on placement task using robotic arm | – No significant difference between 3D MS and 3D SS; 2D display outperformed both 3D displays |
| Richard et al. [42] | 2 studies: SS vs. MS | Efficiency in estimating virtual distances (using haptic glove) | – In baseline conditions, users were more efficient with SS than MS<br>– With high FR and multimodal cues, however, the displays yielded similar performances |
| Scribner and Gombash [47] | SS vs. MS | Errors, efficiency, stress, & usability on UAV navigation task | – SS resulted in fewer errors, reduced stress scores, and was preferred by users (usability) over MS |

Table A-6: Summary of studies manipulating environmental detail [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Chen and Joyner [56] | Dense vs. sparse targeting area | Targeting errors | – Errors increased with more distractor objects around the target<br>– In difficult conditions, manual control outperformed semi-autonomy |
| Darken and Cervik [61] | Ocean vs. urban virtual environments | Efficiency in navigation | – Users had stronger performance in visually sparse ocean environments than in complex urban environments, regardless of the type of camera |
| Fisher et al. [60] | Display image color (color vs. grayscale) | Efficiency, accuracy | – Color image enabled greater efficiency and increased accuracy for target identification compared to grayscale |
| Folds and Gerth [62] | Dense vs. sparse targeting area | RT to identify new threat in in virtual tracking task | – RT to emerging threat was slower in dense environment<br>– Auditory warnings improved RT more so in dense environments |
| Hardin and Goodrich [57] | 200 vs. 400 Distractor targets | Efficiency and errors in VE search and rescue | – # of distractors had a significant effect on efficiency, but not on errors<br>– Introducing autonomy did not mitigate this impact |
| Murray [136] | Target images were complex vs. simple | Efficiency in monitoring and tracking targets in VE | – Increasing image complexity increased target detection time<br>- Automated mobility improved user performance in complex stimuli conditions |
| Schipani [25] | Difficult vs. easy terrain | Workload ratings of UGV navigation | – Workload increased with greater terrain complexity, whereas platform speed and line of sight with the operator did not impact workload |
| Sellner et al. [138] | Simple vs. complex display images | Efficiency and errors on task decision-making (on stimuli) | – Simple displays decreased decision time, but also increased errors<br>– Integrative presentations reduced the time penalty in complex displays |
| Witmer and Kline [58] | Dense vs. sparse virtual environment | Errors in distance estimation for Virtual environment | – More complex environments did not impact virtual distance estimation |
| Yeh and Wickens [59] | Dense vs. sparse virtual environment | Errors, workload, and trust on target detection | – Users had better performance with low (vs. high) environmental detail<br>– With reliably cued targets, the impact of visual detail was reduced |

Table A-7: Summary of studies manipulating task performance standards [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Cosenzo et al. [72] | # Of targets to photo | Errors in targeting, RT to navigational decisions | – As # targets increased, targeting errors and reaction time to navigational stimuli increased |
| Draper et al. [73] | # Of alerts needing responses | Errors and reaction time in responding to UAV alerts | – Performance degraded as system alerts were more frequent; no interaction between condition and form of responses (manual vs. verbal) |
| Galster et al. [71] | # Of targets to process | Errors, efficiency, and workload in processing targets | – Workload differences emerged favoring the low target condition<br>– 4 UAVs yielded better performance with more targets than 6 or 8 UAVs |
| Mosier et al. [74] | Low or high levels of time pressure | Errors and efficiency in diagnosing system problem | – Adding time pressure increased pilot efficiency, but also increased diagnosis errors; this was worsened by system information conflicts |
| Park and Woldstad [54] | Size of destination for Placement | Efficiency and workload in object transfer with robotic arm | – Less efficiency and higher workload in conditions with smaller targets<br>– 3D displays helped performance in with small targets |
| Schipani [25] | Navigation distance | Workload ratings in VE navigation | – Workload increased with greater distance to travel<br>– Line of sight with the operator did not impact workload |
| Wang et al. [139] | Robot coordination demands | Region explored, victims located, and coord. demands | – Tasks with fewer coordination demands yielded higher productivity<br>– The level of coordination demands varied by the type of robot used |
| Wang et al. [140] | # Of tasks assigned | Victims saved, area explored, efficiency, and workload in search and rescue task | – Users covered more surface area, switched between robots more frequently, and reported less workload with simple exploration task<br>– Users with search and locate tasks had worst production, but this was mitigated with control of 8 UGVs (vs. 4 or 12 UGVs) |
| Watson et al. [40] | Distance in 3-D placement | Errors, efficiency, and usability on virtual object placement | – Placement errors increased with greater distances in addition to task completion time; poor FR worsened this effect |

Table A-8: Summary of studies manipulating the number of robots [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Adams [76] | 1 vs. 2 vs. 4 UGVs | # Of actions, efficiency, and workload for search and transfer | – Slight differences between 1 and 2 UGVs, but efficiency and perceived workload were worse with 4 robots |
| Chadwick [77] | 1 vs. 2 UGVs | Errors and perceived workload in targeting, and navigation | – No significant differences between groups |
| Chadwick [82] | 1 vs. 2 vs. 4 UGVs | RT in target responding and correction | – RT was similar between 1 and 2 UGVs but degraded from 2 to 4 UGVs |
| Chen et al. [127] | 1 vs. 3 UGV and/or UAVs | Errors, efficiency, SA, and workload in targeting | – Targeting errors were equal between 3 platforms and single UAV or UGV, but perceived workload and efficiency suffered |
| Crandall et al. [79] | 2 vs. 4 vs. 6 vs. 8 UGVs for team | Errors and efficiency in navigation and target detection/transfer | – 4 and 2 UGV conditions exhibited fewest lost robots<br>– 6 and 8 UGV condition yielded highest # of target successes |
| Hill and Bodt [78] | 1 vs. 2 UGVs | Perceived workload in navigation and image processing | – Perceived workload was higher with 2 UGVs<br>– Operators reported different levels of impact from adding a robot |
| Lif et al. [80] | 1 vs. 2 vs. 3 UGVs | Efficiency in navigation (# of waypoints) | – 2 or 3 UGVs had equal efficiency (# of waypoints) than 1 UGV |
| Parasuraman et al. [63] | 4 vs. 8 UGVs | Completion time for game, # of games won, workload | – Completion time and win rate deteriorated from 4 to 8 UGVs<br>– As workload increased, automation features had a greater impact |
| Trouvain and Wolf [141] | 2 vs. 4 vs. 8 UGVs | Efficiency and perceived workload in navigation and target processing | – Users performed more overall inspections with 4 and 8 UGVs, but also had more idling time and efficiency loss |
| Trouvain et al. [142] | 1 vs. 2 vs. 4 UGVs | Errors and efficiency in navigation | – Users of 1 UGV had optimal navigation performance<br>– 2 and 4 UGV users were equal in performance |
| Wang et al. [143] | 4 vs. 8. vs. 12 UGVs | Victims saved, area explored, efficiency, and workload in search and rescue task | – Use of 8 UGVs provided optimal production, though effect strength was affected by # of tasks assigned (more tasks yielded a stronger effect)<br>– Users of 4 UGVs reported low workload but also had little production |

Table A-9: Summary of studies examining level of autonomy (LOA) [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Chen et al. [56] | Manual UGV control vs. semi-autonomy (monitor UGV actions) | Targeting errors | – Users performed gunnery tasks in addition to teleoperation<br>– Manual control improved robot task performance over semi autonomy, but at the expense of gunnery task performance |
| Endsley et al. [86] | Ten LOAs in monitoring, generating, selecting, and implementing between human operator and automated system | Efficiency and errors in decision-making | – LOAs which combine human generation of options and automated implementation produced superior results<br>-Joint decision making (human/system collaboration) was detrimental to performance |
| Hardin et al. [57] | Search and rescue mission with with varying levels of autonomy: adaptive, adjustable, or mixed initiative | Efficiency and workload | – Mixed initiative (MI), where operator and UGVs jointly decide on LOA for situation performed better than operator in complete control (adjustable) and complete UGV control (adaptive) |
| Kaber et al. [144] | 5 LOAs and 5 schedules of automation<br>(automation on, then off for a specified time) | Errors, workload, and SA in system control task (decision-making and targeting) | – When automation was cycled on and off, performance was best when the human operator implemented a corresponding strategy<br>– Workload correlated with secondary task performance |
| Kaber et al. [88] | 5 LOAs range from simple support to full automation | Errors, efficiency, workload,<br>and SA | – Increased automation led to performance improvements and reduces subjective workload, but also reduced SA for some system functions |
| Krotkov et al. [145] | None, veto-only (e.g., to avoid damage), or semi-autonomous aid (adjusts course) | Usability in UGV navigation | – Users struggled to adapt strategies around autonomous agent control and steering/navigation trouble may arise if the operator is unable to adjust |
| Luck et al. [89] | 3 LOAs: manual control, veto-only, and autonomous waypoint navigation | Errors, efficiency, and usability for UGV search and rescue | – Increased automation led to performance improvements in both errors and time as well as a buffer from the negative effects of control latency |
| Schermerhorn and Schultz [146] | Exploration/search task with autonomous or non-autonomous robot | Efficiency and satisfaction | – With autonomous robot participants were more accurate, but not faster<br>– Participants seemed to ignore ‘‘disobedience” and preferred working with the autonomous vs. normal robot |
| Wang and Lewis [147] | 3 levels of LOA for team of 3 UGVs: full autonomy, mixed control, full control | Efficiency and usability for UGV search and rescue | – With multiple UGVs, mixed control paradigm (manual control and cooperative automation) provided best performance<br>– Switching attention between robots more frequently performed better in manual and mixed control scenarios |

Table A-10: Summary of studies examining automated aid reliability [27].

| Study | Manipulation | Criteria | Results |
|---|---|---|---|
| Dixon and Wickens [91] | Automated alerts were100% reliable, 67% with false alarms, and 67% with misses | Errors, RT, and SA in UAV targeting and system monitoring | – False-alarm prone automation decreased the use of aids encouraged operators to ignore raw data<br>– Imperfect automation led to better detection of a target miss |
| Goodrich et al. [148] | Manual robot teleoperation vs. semiautonomous navigation via waypoints with or without failure warning | Reaction time | – Autonomy results in less idle time to recognize problems, but without automation aid, this benefit turns into a major obstacle<br>– Automation led to dependence when engaged in secondary tasks |
| Kaber et al. [88] | Normal operation vs. unexpected automation failure | Errors, efficiency, workload, and SA for systems control and decision-making | – In automation failure, lower level LOAs with more human control resulted in the best performance due to increased SA |
| Levinthal and Wickens [83] | No automation, 90% reliable, 60% reliable but prone to false alarms, or 60% reliable but prone to true misses | Efficiency in UAV navigation, RT to system alerts | – Aids prone to false alarms were inhibited performance more than 90% reliable or 60% reliable aids prone to misses |
| Meyer et al. [149] | Automated cuing agent for: 45% vs. 80%<br>reliable; High vs. low overall automation | Errors in quality control decision-making task | – Higher levels of automation resulted in more reliance on cues<br>– No performance differences between LOA conditions for valid cues, but low LOA outperformed high LOA for unreliable cues |
| Rovira et al. [93] | 60% vs. 80% decision reliability in automation aid | Errors, RT, workload, and trust on command and control decision-making task | – Imperfect decision-making automation was detrimental to performance, explained by operator complacency with automation and lack of access to raw data |
| Ruff et al. [150] | 95% or 100% accurate automated or by consent decision-making aid | Errors and workload for UAV targeting and decisions | – Management-by-consent automation aid resulted in best performance as it left operators in the loop but scalable to increases in workload (more UAVs) |
| Wickens et al. [151] | Automated diagnostics information: none,<br>100% accurate, 60% reliable w/false-alarms, 60% reliable w/misses | Errors and efficiency for UAV navigation, targeting, systems monitoring | – Automation prone to misses decreased concurrent task performance, whereas automation prone to false alarms led to slower RT to all auto-alerts and decreased efficiency, accuracy |
| Yeh and Wickens [59] | 75% vs. 100% reliable cuing for some targets | Errors, workload, and trust on UAV targeting | – Partially reliable cuing increases false alarms and eliminates overall performance benefits of cuing; Cuing draws attention towards cued target results in other targets being overlooked |

## APPENDIX B

## DEMOGRAPHIC SURVEY

Participant #: Age: Gender: Male / Female Date:

1. How often do you

- Drive a car?

  Daily Weekly Monthly

  Once every few months Rarely Never

- Use a joystick/steering wheel?

  Daily Weekly Monthly

  Once every few months Rarely Never

- Play computer/video games?

  Daily Weekly Monthly

  Once every few months Rarely Never

2. Which type(s) of computer/video games do you most often play if you play at least once every few months?

3. Which of the following best describes your expertise with computer? (Check one)
_____ Novice
_____ Good with one type of software package (such as word processing or slides)
_____ Good with several software packages
_____ Can program in one language and use several software packages
_____ Can program in several languages and use several software packages

4. Are you in your usual state of health physically? YES NO
   If NO, please briefly explain:
5. How many hours of sleep did you get last night? _____ hours
6. Do you have normal color vision? YES NO
7. Do you have prior military service? YES NO If Yes, how long _____

## APPENDIX C

## NASA-TLX QUESTIONNAIRE

Please rate your **overall** impression of demands imposed on you during the exercise.

1. Mental Demand: How much mental and perceptual activity was required (e.g., thinking, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

LOW |---|---|---|---|---|---|---|---|---|---| HIGH

1 2 3 4 5 6 7 8 9 10

2. Physical Demand: How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

LOW |---|---|---|---|---|---|---|---|---|---| HIGH

1 2 3 4 5 6 7 8 9 10

3. Temporal Demand: How much time pressure did you feel due to the rate or pace at which the task or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

LOW |---|---|---|---|---|---|---|---|---|---| HIGH

1 2 3 4 5 6 7 8 9 10

4. Level of Effort: How hard did you have to work (mentally and physically) to accomplish your level of performance?

LOW |---|---|---|---|---|---|---|---|---|---| HIGH

1 2 3 4 5 6 7 8 9 10

5. Level of Frustration: How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

LOW |---|---|---|---|---|---|---|---|---|---| HIGH

1 2 3 4 5 6 7 8 9 10

6. Performance: How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

LOW |---|---|---|---|---|---|---|---|---|---| HIGH

1 2 3 4 5 6 7 8 9 10