CHALMERS
UNIVERSITY OF TECHNOLOGY

Teleoperation of Autonomous Vehicle
With 360° Camera Feedback

Master's thesis in Systems, Control and Mechatronics

OSCAR BODELL
ERIK GULLIKSSON

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

MASTER'S THESIS IN SYSTEMS, CONTROL AND MECHATRONICS

Teleoperation of Autonomous Vehicle

With 360° Camera Feedback

OSCAR BODELL
ERIK GULLIKSSON

Department of Signals and Systems
Division of Control, Automation and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Teleoperation of Autonomous Vehicle
With 360° Camera Feedback
OSCAR BODELL
ERIK GULLIKSSON

Master's Thesis 2016:43
Department of Signals and Systems
Division of Control, Automation and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 (0)31 772 1000

Cover: Teleoperation setup with stitched camera view & graphical user interface.

Chalmers Reproservice
Gothenburg, Sweden 2016

Teleoperation of Autonomous Vehicle
With 360° Camera Feedback
Master's thesis in Systems, Control and Mechatronics
OSCAR BODELL
ERIK GULLIKSSON
Department of Signals and Systems
Division of Control, Automation and Mechatronics
Chalmers University of Technology

Abstract

Teleoperation is using remote control from outside line of sight. The operator is often assisted by cameras to emulate sitting in the vehicle. In this report a system for teleoperation of an autonomous Volvo FMX truck is specified, designed, implemented and evaluated. First a survey of existing solutions on the market is conducted finding that there are a few autonomous and teleoperation solutions available, but they are still new and information is sparse. To identify what types of requirements are needed for such a system and how to design it a literature study is performed. The system is then designed from the set requirements in a modular fashion using the Robot Operating System as the underlying framework. Four cameras are mounted on the cab and in software the images are stitched together into one 360° image that the operator can pan around in.

The system is designed so that the operator at any time can pause the autonomous navigation and manually control the vehicle via teleoperation. When the operators intervention is completed the truck can resume autonomous navigation. A solution for synchronization between manual and autonomous mode is specified. The truck is implemented in a simulation where the functionality and requirements of the system is evaluated by a group of test subjects driving a test track.

Results from simulation show that latencies higher than 300 ms lead to difficulties when driving, but having a high frame rate is not as critical. The benefit of a full 360° camera view compared to a number of fixed cameras in strategic places is not obvious. The use of a head mounted display together with the 360° video would be of interest to investigate further.

Keywords: Teleoperation, autonomous vehicle, surround view, remote steering.

Acknowledgements

First of all we would like to thank CPAC Systems for letting us become part of the family and our supervisors at CPAC Annegret Render and Michael Pettersson. Thank you also Knut Åkesson for being our supervisor and examiner at Chalmers University of Technology and helping us with the report. A final big thank you to Josef Kindberg for your help and expertise in ROS.

Oscar Bodell, Erik Gulliksson, Gothenburg, June 2016

Contents

Nomenclature

${}^{*}B$	Nr. of **B**ytes
${}^{*}b$	Nr. of **b**its
5G	**5**th **G**eneration Wireless Systems
AHS	**A**utonomous **H**aulage **S**ystem
CAN	**C**ontroller **A**rea **N**etwork
CPU	**C**entral **P**rocessing **U**nit
FOV	**F**ield **O**f **V**iew
FPS	**F**rames **P**er **S**econd
FSPL	**F**ree-**S**pace **P**ath **L**oss
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem (see GNSS)
GPU	**G**raphics **P**rocessing **U**nit
GUI	**G**raphical **U**ser **I**nterface
HMD	**H**ead **M**ounted **D**isplay
HSV	**H**ue **S**aturation **V**alue
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IMU	**I**nertial **M**easurement **U**nit
IP	**I**nternet **P**rotocol
LAN	**L**ocal **A**rea **N**etwork
LiDAR	**Li**ght **D**etection **A**nd **R**anging
LOS	**L**ine **O**f **S**ight
OEM	**O**riginal **E**quipment **M**anufacturer
OpenCV	**Open** source **C**omputer **V**ision library
PoE	**P**ower **o**ver **E**thernet
RGB	**R**ed **G**reen **B**lue
ROI	**R**egion **O**f **I**nterest
ROS	**R**obot **O**perating **S**ystem
RTK	**R**eal **T**ime **K**inematic
RTSP	**R**eal **T**ime **S**treaming **P**rotocol
RTT	**R**ound **T**rip **T**ime
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
Volvo FMX	**V**olvo **F**orward control **M**edium **X**treme
VR	**V**irtual **R**eality
Wi-Fi	IEEE 802.11xx wireless communication
WLAN	**W**ireless **L**ocal **A**rea **N**etwork

List of Figures

List of Tables

1 Introduction

Autonomous vehicles is one of the more exciting areas of the automotive industry today. The general perception is a fully automated car where the driver can handle other matters while the car is driving by itself and this is starting to become a reality with passenger cars. However construction machines and industrial vehicles are different. In many cases their purpose is not to transport the driver from one point to another as in a car but instead perform tasks at a work site. Even if the equipment will be able to carry out the work without support from an operator there is a need for supervision and to be able to take control remotely if something unexpected happens. For this to function a system has to be designed, configured and built as a control center. In this control center an operator will be able to supervise the vehicle and monitor the status of the vehicle. If needed the operator can take control and give the vehicle appropriate commands for it to be able to continue its task.

Consequently, in autonomous operation, no driver is there to supervise and operate the vehicle if something goes wrong. An example of a scenario is when a vehicle gets stuck behind an obstacle and cannot find its way around it. Instead of deploying an operator to go to the vehicle and drive it, this can be done remotely. This is in many cases safer and more efficient. Therefore teleoperation is an helpful tool before the vehicles are fully autonomous and can handle all types of obstacles on their own.

The work will focus towards a generic solution that can be scaled and utilized on different vehicles for several applications. The design will be flexible and the system will be implemented towards an all-terrain truck where it will be tested and evaluated.

1.1 Purpose & Objective

The purpose of this thesis is to specify requirements for, design and implement a prototype of a system for teleoperation of a normally autonomous vehicle. The existence of standards will be investigated. If present, the standards will be adhered in the development and implementation of the system. In any case a general and scalable solution that can be used on several types of vehicles will be developed. An interface towards the autonomous vehicle will be created together with a control center with controls and information from the vehicle. The system can be used when the vehicle cannot navigate autonomously or is in a situation when it is more convenient and/or safe to operate the vehicle remotely.

1.2 Main Research Questions

In order to fulfill the purpose, the following questions will be answered:

- How shall camera images, maps and sensor data be presented in order to maximize the safety and efficiency of the operation?
- At what level does the operator control the vehicle? As if sitting inside or are more high level commands (i.e. "Go to unloading location") issued? How do delays in the communication channel affect the choice of control?
- Are there existing standards for remote control of autonomous vehicles?
- How can the system be scalable to a variety of different sensors depending on application, and what are the requirements of the communication link for different types of sensors?
- How will the vehicle switch between autonomous operation and manual control? What system has priority in different situations, what kind of handshakes are needed?

1.3 Boundaries

The communication link from the control center to the vehicle itself will not be implemented, but requirements will be specified. No autonomous functions will be developed, those are assumed to already exist in the vehicle. Maps used for autonomous navigation and presentation to the user are assumed to exist everywhere the vehicle is driven.

For teleoperation and autonomous control, only vehicles will be investigated since they operate in a similar fashion, often with steering wheel and pedals or joysticks. The implementation and evaluation will be carried out on an all-terrain truck with no tasks other than transporting and unloading goods.

The work will be carried out during 20 weeks in the spring of 2016 on readily available hardware. Due to of the limited time frame, open source solutions such as ROS and OpenCV will be used to speed up the development.

1.4 Method

Initially a literature study of teleoperation of autonomous work vehicles and a market survey of existing solutions (see *3 - Existing Solutions and Standards*) were performed to gain knowledge of the different parts and aspects in the system and what insights can be gained from previous solutions. Several manufacturers have models that drive autonomously and are expanding to more models and features. However most projects are still in development stage and tests. Theory regarding latency, video feedback and additional sensor information that is relevant to the operator has been gathered including different type of communication technologies to transfer commands and sensor data.

In order to build a prototype system a certain number of requirements have to be specified to aid the system design. These requirements can be found in *4 - System Requirements* where each requirement is prioritized from one to three depending on the necessity in system design, and it is also specified how the requirement will be evaluated.

The prototype system is divided into two parts, the control center and the actual vehicle which is presented in *5.1 - System Architecture* together with presentation of each subsystem in *5.3 - Subsystems*. The choice of dividing the system into smaller subsystems, is to make the solution flexible and scalable since different subsystems can be added or removed due to different applications and sensors available.

Evaluation of the system and its subparts is done in a simulation environment described in section *5.5 - Gazebo Simulation* . In this environment it is possible to test each part of the system and evaluate against the requirements. The simulation is also used to evaluate if the supporting functions are beneficial for the operator together with complete system design evaluation. The ability to make changes to the system and measure the affect in performance by increasing latency, varying quality in the video feed, using different support functions and limiting the types of control input is implemented.

The results gained from the evaluation is then compared to the specified requirements if these are met or not in terms of both driving experience and system design in *6 - Results* and *6.3 - Evaluation*. From the results conclusions are drawn on how implementation of teleoperation in an already autonomous vehicle shall be implemented and experience gained from the evaluations and tests. This together with thoughts on future work is presented in *7 - Conclusion & Future Work*.

1.5 Thesis Contribution

There are several autonomous or teleoperated work vehicle solutions today. However the solutions are often implemented on a specific vehicle type from the original equipment manufacturers and retrofit solutions typical lack the ability to control autonomously. Therefore an integrated system is proposed where both autonomous and teleoperated technologies are combined into one control center for monitoring and control of autonomous vehicles. The ability to pause an ongoing autonomous mission and manually control the vehicle and then resume the mission is an important function. The system is scalable and flexible depending on the type vehicle and application. The ability to define new autonomous tasks or sequences while driving the vehicle remotely has not been seen in any other solution today which is a feature that will be beneficial for the operator. The stored autonomous tasks could be driving a path as in this case, but also control of equipment etc. Different operator assists are evaluated to assess which ones that are important for the operator to maneuver the vehicle safely and precise.

Existing solutions for teleoperation use multiple fixed cameras that the user can choose from. Switching between different cameras causes the operator to have to re-orient from the new point of view. The proposed system uses a 360° video image that the operator can pan in, as if looking around in real life. This is expected to improve telepresence. Variations in frame rate and latency are explored in order to investigate how much is acceptable for a safe and efficient operation.

1.6 Thesis Outline

This thesis paper is divided into seven chapters. The chapter *1 - Introduction* is followed by chapter *2 - Application Overview*. It briefly describes the setup and some key features. Further *3 - Existing Solutions and Standards* follows where the results of the a market survey is presented. The specified *4 - System Requirements* are then presented with background and evaluation method. Then the *5 - System Design* is described first with system architecture followed by the subsystems and ends with simulation set-up. This is followed by *6 - Results* where the results from the simulation are given and lastly the conclusions are stated in *7 - Conclusion & Future Work*.

2
Application Overview

The proposed system in this paper is designed to be applicable to a variety of different vehicles and machines. However it will be implemented and tested as an all terrain haulage truck used in a known closed off environment. The aim of the system is to be able to control the vehicle without being in the physical vicinity of it. This will be done by relaying information to the operator from the vehicle such as video streams and sensor data. The operator will be able to send control inputs to the vehicle in order to maneuver it. The presented implementation consists of functionality and software that can be used for teleoperation control and can be run on a regular personal computer. The primary purpose is to evaluate the requirements set and help answer the research questions stated in introduction. Hence it is not a final control center ready for commercialisation.

2.1 Level of Autonomy

There are many ways of controlling a vehicle, but the most common way is still with the operator sitting inside the vehicle driving it manually. Other ways are remote and autonomous control, and these technologies are often divided into three levels of control. The first is manual remote control [1] which contains no autonomous functions. The operator controls the vehicle from a near distance where the vehicle can be viewed directly while operated. This is often referred to as line of sight control.

The next level is teleoperation where the operator is located off site and some sort of monitoring is needed i.e. cameras, force feedback control or other sensor data. Teleoperation [2] can both be local where the operator is located close to the machine but not in visible range. It can also be global where communication needs to be relayed via the Internet or by a satellite link. Different kinds of autonomous tasks can be used by the operator at this level.

The third step is a fully autonomous vehicle [3] that can carry out tasks on its own with no guidance of an operator. The requirements are significantly higher at this level in terms of positioning, operation and safety. The tasks can be predefined and depending on situation the vehicle must be able to make its own decisions [4].

2.2 Evaluation vehicle

The vehicle that is used for the prototype is a Volvo FMX [5] construction truck equipped with a variety of additional sensors such as an IMU (Inertial Measurement Unit) to measure orientation, LiDAR (Light Detection And Ranging) sensors to measure distance to surroundings and centimeter precision positioning using RTK-GNSS (Real Time Kinematic - Global Navigation Satellite System). Details about the sensors can be viewed in section *5.3.5 - Additional Sensors* . The truck has autonomous capabilities implemented and can follow a pre-recorded path with the position, orientation and desired speed of the truck at discrete waypoints along the path, called *bread crumbs* [6]. As of right now there exists no other path planning except manually driving and recording a path. Actuators and interfaces for steering and controlling brake and throttle are available. The vehicle is implemented in the simulation software Gazebo (see section *5.5 - Gazebo Simulation*) to be used for the evaluation, a screenshot can be seen in Figure 2.1.

Figure 2.1: Screenshot of the evaluation vehicle in the Gazebo simulation.

2.3 Cameras and stitching

In addition to the sensors already mounted on the truck, it is equipped with a number of cameras mounted so that a full surround view from the truck will be achieved. These camera images are then stitched together to a single image containing all camera streams. The operator will then be able to pan around in this image in order to emulate looking around while sitting in the vehicle.

2.4 Visualization and Operator Support

The stitched camera feed in this prototype is shown in a window on the computer running the system. On top of the video, relevant information for the operator is overlaid. These can be a map, vehicle speed, vehicle width markings and other types of support. When the user pans in the video feed, the overlaid information stays in place, but the video below will move. Using the GNSS data the position and heading of the vehicle is displayed in the map. The range information from the LiDARs is integrated so that unknown obstacles are shown in the map. The whole image can be seen in Figure 2.2

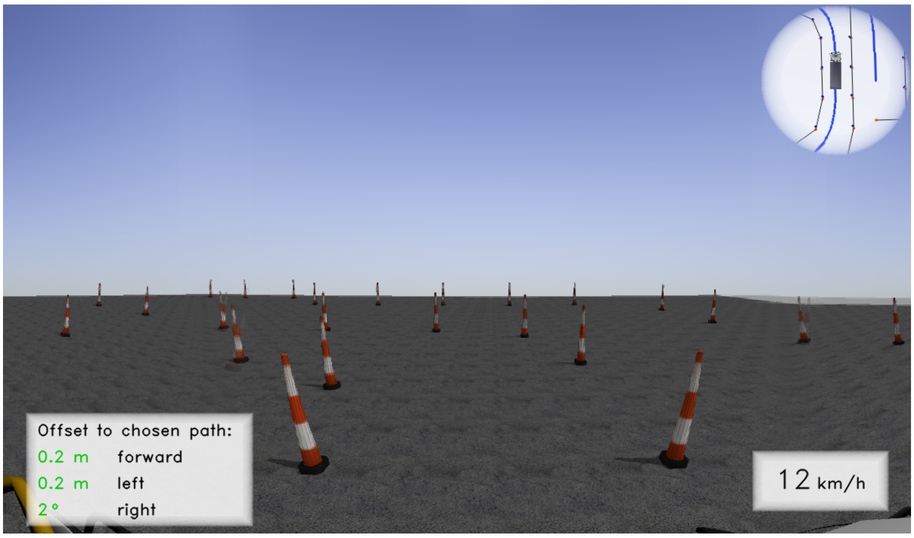


Figure 2.2: Screenshot from the stitched video feed overlaid with map, speedometer and align assistance for autonomous paths.

2.5 User Controls

This prototype system uses multiple control inputs to evaluate different types of controls. Simple consumer inputs are used such as a steering wheel or a gamepad normally used for computer games. In addition to driving the vehicle, buttons are used to control operator support functions used while driving, such as zooming in a map or panning in the video feed. A simple user interface is present for the operator to change settings for the maps and autonomous functions. In addition it shows vehicle information together with a map. The user interface can be seen in Figure 2.3

2.6 Autonomous Functions

The truck can follow pre-recorded paths autonomously. When in autonomous mode the truck will stop for obstacles using the data from the LiDAR sensors. The truck will then wait until the obstacle disappears. One of the primary purposes of this teleoperation system is to control the vehicle when it cannot navigate autonomously. That could be when it has stopped in front of an stationary obstacle. Therefore the system can interrupt the autonomous navigation and take control over the vehicle so the operator can drive manually. When the manual override is done, the operator

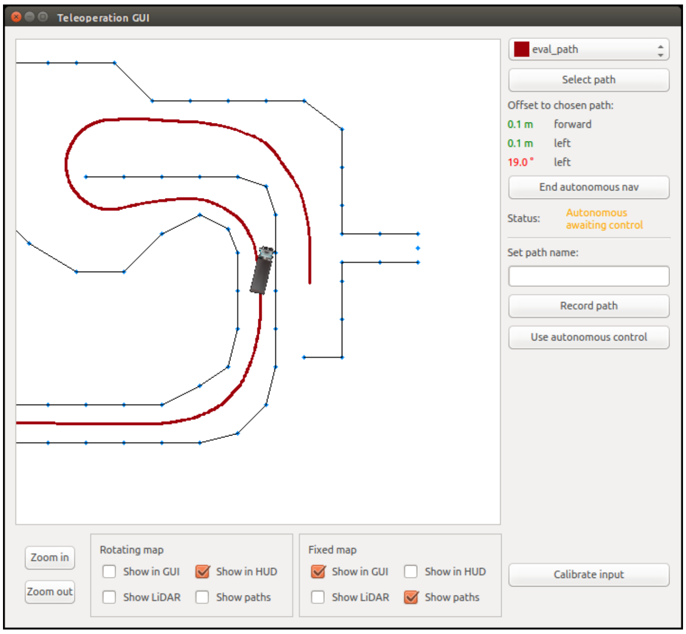


Figure 2.3: Screenshot from the operator interface controlling autonomous functions and maps.

will have to stop the vehicle somewhere on the path that the vehicle is following. Then autonomous navigation can be resumed.

2.7 Implementation

The implementation of the system will be modular with multiple subsystems. This is done using Robot Operating System (ROS), a framework already present on the truck. ROS simplifies communication of subsystems to create a larger robust system. To evaluate the impact of latency, varying frame rates and the different operator support functions the prototype is tested in a simulation.

This proposed system is a mixture of level two and three of autonomy. The existing technology is already autonomous, and teleoperation utilizing cameras and the available sensors is added. The transition between autonomous and teleoperation has to be specified and implemented.

3

Existing Solutions and Standards

In order to gain insight and conform to standards a literature study and a market survey was conducted. The survey was conducted by investigating the solutions offered in the market today mainly by the information given by respective manufacturers website together with press releases and articles. Since most of the investigated solutions are new, information about the systems and the performance is limited.

The survey is divided into three parts. First the findings from a study of existing standards that apply for this prototype is presented. Then remote control systems integrated into the vehicle from the manufacturer called Original Equipment Manufacturer (OEM) solutions are presented. Following are aftermarket or retrofit solutions where existing equipment is augmented with third party technology. Each solution is based on either of the following three categories or a combination of them; Line Of Sight (LOS) remote control, teleoperation or fully autonomous functionality.

3.1 Existing standards for remote control

Using standards allows for a more unified market where accessories are compatible with different platforms and equipment from several manufacturers can work together. But by creating a closed ecosystem the manufacturer can sell their own products or products from selected partners. The standards relevant for this project are standards that dictate how to send commands to an autonomous vehicle or pause an ongoing autonomous mission. Literature and the main standard associations (ISO, IEEE, ANSI, IEC etc) were surveyed but standards for this specific application has not been developed yet. Standards exist for testing this type of product ready for production, but since this is an early prototype it is not applicable.

3.2 Original Equipment Manufacturer Solutions

A number of OEM solutions have been examined with the following manufacturers; Caterpillar, Komatsu, Hitachi/Wenco, Sandvik and Atlas Copco. All these companies have a complete solution on the market and further implementations are undergoing. The majority of these implementations are in-house solutions that only work with the manufacturer or specific partners' vehicles and machines. The results from the OEM survey can be viewed in Table 3.1 and 3.2.

Table 3.1: OEM solutions table 1 of 2

	Caterpillar	Caterpillar	Caterpillar	Komatsu	Hitachi/Wenco
Model/Type	D10T/D11T[7]	793F[8]	R1700G[9]	AHS[1] 930E/830E[10]	AHS[1][11]
Vehicle/Equipment	Bulldozer	Mining Trucks	Wheel Loader	Mining Trucks	Mining Trucks
Operation Area	Surface Level	Surface Level	Underground	Surface Level	Surface Level
LOS Remote	Yes	N/A	No	Yes	Yes
Teleoperation	Yes	N/A	Yes	Yes	Yes
Autnonomous	No	Yes	Semi	Semi	In Development
Multiple Vehicles	N/A	Yes, Cat Minestar	N/A	Yes	Yes
Communication	Radio 0.9/2.4 GHz	E-Stop at 919 MHz	WiFi	N/A	N/A

Table 3.2: OEM solutions table 2 of 2

	Sandvik AutoMine	Sandvik AutoMine	Sandvik AutoMine	Atlas Copco	Atlas Copco
Model/Type	AHS[1][12]	Loading[13]	Surface Drilling[14]	Scooptram[15]	Benchremote[16] SmartROC D65
Vehicle/Equipment	Trucks	Loaders	Drill Mining	Wheel Loader	Drilling
Operation Area	Underground	Underground	Surface Level	Underground	Surface Level/ Underground
LOS Remote	N/A	Yes	Yes	Yes	Yes
Teleoperation	N/A	Yes	Yes	Yes	No
Autnonomous	N/A	Semi	No	Semi	Yes
Multiple Vehicles	Yes	Yes	Yes	N/A	Yes, up to 3
Communication	N/A	N/A	N/A	Bluetooth/WiFi	WiFI

Caterpillar's Minestar system [17] is a complete system for mining activities from monitoring, diagnosing, detection and command. The system is scalable to fit different needs and expandable for development. Komatsu [10] has a similar system as Caterpillar's. They both function by sending certain commands for final position and speed, and the trucks will navigate autonomously. Positioning is done using GNSS which requires that the tasks are performed above ground. Hitachi/Wenco are developing a similar autonomous haulage system [11] that it is to be launched 2017.

AutoMine is a system developed by Sandvik [12] which is one of the world's leading companies in automation of mining operations. AutoMine consists of mainly three different parts; AHS[1], loading and surface drilling. The AHS works similar to previously described competitors Cat and Komatsu. The AutoMine Loading can be controlled by teleoperation and has the ability to drive autonomous when transporting the load. The operator can therefore handle multiple loaders simultaneously. AutoMine surface drilling is a remote controlled drilling machine solution that can be operated from both local and global sites. Multiple drills can be operated simultaneously by one operator.

Atlas Copco has a similar underground loading solution as Sandvik with their Scooptram [15]. The loading is done by teleoperation but transportation can be done

[1] Autonomous Haulage System

autonomously. In addition Atlas Copco has an operating station for remote control of drilling machines [16]. Each station can handle up to three different drilling machines simultaneously, but the station has to have free line of sight in order to function.

3.3 Retrofit Solutions

There exists several after market solutions for remote control of vehicles and machines. Most of the systems use line of sight remote control but some offer complete solutions for autonomous driving and monitoring. For most of the solutions the operation needs to be located at surface level since the use of GNSS. The results from the retrofit survey can be viewed in Table 3.3 and 3.4.

Table 3.3: Retrofit solutions table 1 of 2

	Remquip	**ASI Robotics**	**Hard-Line**	**TorcRobotics**
Model/Type	[18]	Mobius, NAV[19]	[20]	[21]
Vehicle/Equipment	Hydralic Machines	Mining, Trucks Cars, etc..	Construction Vehicles	Construction Vehicles
Operation Area	Surface Level	Surface Level	Surface Level/ Underground	Surface Level
LOS Remote	Yes	Yes	Yes	Yes
Teleoperation	No	Yes	Yes	Yes
Autnonomous	No	Yes	No	Semi
Multiple Vehicles	No	Yes	No	N/A
Communication	Radio	N/A	Radio/WiFi	N/A

Table 3.4: Retrofit solutions table 2 of 2

	AEE	**Taurob**	**Oryx Simulations**
Model/Type	[22]	Universal Teleoperation Control[23]	Interface for Teleoperation[24]
Vehicle/Equipment	Smaller Construction Machines	Construction Machines, Trucks	3D Simulators
Operation Area	Surface Level	N/A	Surface Level/ Underground
LOS Remote	Yes	Yes	N/A
Teleoperation	Yes	Yes	N/A
Autnonomous	Yes	No	No
Multiple Vehicles	Yes	No	No
Communication	WiFi	N/A	N/A

The companies most relevant to the project are ASI Robotics (*Autonomous Solutions, Inc*) and AEE (*Autonomous Earthmoving Equipment*). Both have solutions for autonomous driving. ASI robotics' solution [19] can be used on several different kinds of vehicles, from ordinary cars to construction and farming machines. Their product is scalable from LOS remote control to autonomous driving of several vehicles with their Mobius and NAV devices. The system is closed so it is difficult to combine with other solutions. AEE can control smaller construction machines autonomously. Similar to ASI the system is scalable from LOS remote control to autonomous control with path planning.

Oryx Simulations does not offer remote control for vehicles but builds 3D simulators [24] for construction vehicles. It is therefore interesting how the cab interface has been implemented to achieve a realistic simulation of a real vehicle.

3.4 Outcome of Market Survey

No existing standards has been found on how to achieve teleoperation of autonomous construction vehicles. Several companies and manufacturers have solutions and are expanding to more models and functions. Most of the solutions are proprietary with the effect that choices are limited when it comes to expandability and changeability.

Since most of the products are newly released and under development it is difficult to compare their performance. Most of them have shown that their solutions work and have implemented test vehicles for evaluation. Since the OEMs are large companies with complete solutions it is not very likely that they will collaborate to create a unifying standard in the near future.

4

System Requirements

For the system to behave as intended, a number of requirements have to be specified. What types of requirements are needed, how they influence the system and what the actual requirement is, is described below. If applicable it is also stated how the requirement will be evaluated. The background of the requirements origins from studies of existing literature. These requirements can be viewed in Table 4.1 and have been prioritized depending on the importance for the functionality in the system. Priority *1* is the highest and is set to requirements that are vital for the system to work as intended. Properties with priority *2* are requirements that are to be implemented, but the system would still work as intended without them. Priority *3* are features to expand and enhance the system. These are features that would be interesting to evaluate to see if there is a performance increase.

Some requirements are implemented so that the value can be varied to test if it affects performance of operation. This is done in order to evaluate if the requirements specified are appropriate. Including for instance video latency, frame rate variations or proximity indication that can be enabled or disabled. This is also specified in Table 4.1

4.1 Identification of Requirements

Before creating the different parts of the system described in Chapter *2 - Application Overview*, requirements for each part needs to be specified to achieve a certain performance, driveability and level of safety. Cameras are used to create the surround view of the truck and requirements on a certain field of view and frame rate for the image are set. Relevant information has to be presented to the operator and therefore it is specified what kind of information and how it should be presented, this can include sensor information, maps, vehicle status etc.

Keeping latency or delay time small in the system is of great importance for remote control. A total round trip time from that the operator gives input to the system to the operator gets feedback from video and maps is set. Latencies in the different subsystems would be beneficial to measure for evaluation purposes. This can be done inside ROS since each message has time stamps and ROS uses synchronized time (see section *5.2 - Framework - Robot Operating System*).

Since the vehicle has autonomous functions implemented requirements are needed to make sure that the transition between the teleopration and autonomous mode is in a stable state at all time and also what will happen when the autonomous functionality

Table 4.1: Requirements on the system for teleoperation, priority and how to verify them

Criteria	Value	Variable	Priority	Verification
Autonomous synchronization				
Manual takeover from autonomous			1	**S**
Resume autonomous after manual			1	**S**
Autonomous start	From path start		1	**S**
	Anywhere on path		2	**S**
Autonomous tasks				
Record new paths				
in teleoperation mode			2	**S**
Communication link				
Latency	Max 20 ms	Yes	1	**L**
Data capacity	Min 17 Mbit/s	Yes	1	**C**
Orientation Map				
Fixed map and rotating vehicle	On/off		2	**S**
Rotating map and fixed vehicle	On/off		2	**S**
Representation of LiDAR data	On/off		2	**S**
Sensor data presentation				
Speedometer	Visible		1	**S**
Vehicle attitude	Visible at danger		3	**S**
Distance to obstacles	Visible when close		2	**S**
Proximity warning	Visible when close	On/off	3	**S**
Teleoperation				
Speed limit	30 km/h		1	**S** & **T**
Desired steering angle			1	**S**
Desired acceleration			1	**S**
Desired breaking			1	**S**
Gear box control			2	**S**
Parking brake control			2	**S**
Control types	Steering wheel,		1	**S**
	Gamepad, Joystick		2	**S**
Video				
Latency	max 500 ms	Yes	1	**I**
Frame rate	min 15 FPS	Yes	1	**I**
Field of view	360°		1	**S**
Image quality	Road sign, 15 metres	Yes	2	**T**

T = Live test, **S** = Verify in simulation, **I** = Implement meter, **L** = Measure with *ping*, **C** = Measure with *iperf*

cannot handle a certain situation. The operator should have the ability to abort the autonomous drive and take over by driving manually but also resume paused autonomous tasks.

4.2 Video Feedback

To percept the environment of the vehicle, video is a very important tool. Different ways of presenting the video to the user have effects on the operators ability to handle the vehicle. A narrow field of view makes it difficult for the driver to navigate correctly because there is no peripheral vision where walls, ditches, lines or other objects can be seen. This is known as the "keyhole effect" [25][26]. It has been found [27] that a restricted field of view negatively influences the users ability to estimate depths and the perception of the environment. Because a driver relies on the "tangent point" [28] when driving through a curve it makes it more difficult to navigate through curves with reduced peripheral vision.

A wide field of view can counteract these negative effects, it will be easier for the operator to interpret the surroundings, navigate and control the vehicle. But since the larger image is presented in the same visual space, there is a lack of detail in the image compared to a closer one. The big quantity of visual information makes the perceived speed much higher. This can make the operator drive much slower than needed [29], resulting in an inefficient operation.

The aim for the field of view is to get a complete 360° view around the vehicle. However depending on the presentation to the operator either using a monitor setup or head mounted display (HMD) the presented field may differ. If a HMD is used, the full 360° view will not be displayed but instead the operator will be able to "look around" in 360°. The monitor setup also dictates how much of the image that will be shown. With a smaller monitor it might be better to display a smaller view of the surroundings and to let the user pan, with multiple monitors maybe the whole image can be displayed to create a full view.

The frame rate of the video stream is important to get a smooth experience and enough visual information when viewing the video stream, and it is specified to a minimum of 15 FPS. The frame rate will be measured in the video processing to evaluate if the set requirement is appropriate.

A proposed solution to evaluate the quality of the images is that a human should be visible or that a road speed limit sign should be possible to be read at certain distances. The distance required depends on the travelling speed of the vehicle, the faster the vehicle moves the longer the stopping distance will be. It is here specified to 15 meters. Obstacles needs to be observed early enough to stop the vehicle if necessary.

4.3 Latency Effects

The delay time from the input to the response the operator experiences is known as latency. This is of one the most challenging problems [30][29] to deal with in remote control of vehicles. Depending on the amount of latency it may not even be possible to achieve manual remote control. This is because the system might be unstable if it takes several seconds for the vehicle to respond to the commands from the operator. The video and sensor data which is the response to the operator will be old and therefore incorrect. However humans are able to compensate for delays [30] and instead of making continuous inputs, the operation will turn into a stop and wait scenario when latency reaches about one second. Large delays will therefore impact the safety, operation times and also the performance and efficiency.

Large latencies can induce motion/cyber sickness [31] as the visual effects lags behind reality. High latency will also reduce the perceived telepresence [29], the perception of being present in a virtual environment. In the presence of large latencies, the operator might not be able to see an obstacle emerging suddenly into the trajectory, and thus not being able to avoid it or brake in time. Therefore it is important that there is an automatic emergency braking system [32] in place if the latency is large.

Since is of great importance to keep the delay time low to get good performance, the total round trip time from input controls to video feedback is set to 500 ms and it will be measured through all subsystems in the simulation to achieve the total latency.

4.4 Sensor Data Presentation

Relevant data needs to be presented to the operator and therefore requirements are set to present the speed of the vehicle, attitude, distance to upcoming objects together with proximity warning. This information can be presented either on a separate monitor screen or as head-up information in the video stream. In order not to show unnecessary data to the operator, the attitude of the vehicle together with the distance to objects may only be visible when needed as the vehicle getting close to a dangerous attitude or close to objects and obstacles. Other types of data that is of interest for driving and monitoring the vehicle that needs to be presented could be vehicle fault codes, fuel usage, gear indicator, rpm etc.

4.4.1 Map

A map of the surroundings is needed to display the vehicle together where obstacles and work areas are located. The vehicle is seen from a top-down view where it is either fixed with the map rotating or a fixed map with the vehicle rotating as mentioned in section *5.3.3 - Maps* . The size of the vehicle and distance to near surroundings in the map should be displayed true to scale to give the operator a better intuition of how far the vehicle is from an obstacle.

4.5 Driver Control Inputs

When maneuvering the vehicle in teleoperated mode the natural choice is a steering wheel with throttle and brake pedals in order to mimic sitting in the vehicle. However, evaluating other types of control inputs could show that different types of inputs improves operation such as gamepads and joysticks. Consequently, multiple inputs are required for evaluation in this implementation, more about this can be found in *5.3.4 - Control Inputs*.

4.6 Autonomous Synchronization

The takeover between manual teleoperation and autonomous driving has to be specified. When the vehicle is driving in autonomous mode the operator should be able to take control of the vehicle at any point independent of the state of the vehicle. When in manual mode it should be possible to start autonomous tasks and also resume tasks if interrupted. Autonomous tasks and paths should be able to be defined while driving in teleoperation mode and stored for later use.

The autonomous vehicle follows pre-recorded paths (see section *5.4.2 - Autonomous Functions*). In order to start autonomous navigation the vehicle needs to be stopped on such a path before the autopilot is engaged. The vehicle will then follow the path until it reaches a point on the path specified by the operator or the end of the path and it will stop. If the vehicle is driving autonomously the system will always be able to switch over to manual control. The vehicle will then stop before manual control is granted. A requirement is that the vehicle should be able to resume its autonomous drive after the manual control. This requires the operator to stop the vehicle on the current path and order it to resume.

4.7 Communication Interface

To control the vehicle, interface commands are needed to be transmitted from the control center to the vehicle. These commands have to be specified to meet the system requirements. Essential commands to control the vehicle in both autonomous and teleoperation mode are desired steering angle, throttle and brake. For full maneuverability in teleoperation mode commands for shifting gears are required to be able to reverse together with parking brake commands. More specific commands for the system can be the ability to tip the platform etc. Other useful commands are control of the lights on the truck which includes high beams to use in darkness and turn signal lights to signal the direction in intersections etc. This will require access to the vehicle's CAN (Controller Area Network) interface on the real truck which is the data bus on the vehicle but in the simulation this does not exist.

Status messages from the vehicle to the control center are required to monitor the condition and feedback from the driving. In addition to the messages from the external sensors used, a number of data messages are needed. This can include the actual steering angle, speedometer, rpm and gear indicator. If fault codes are set in the vehicle these need to be forwarded to the operator in order make appropriate actions. Other status messages that may benefit operation are different kinds of status indicators for the vehicle. This can be indicators if high beams are being used, fuel level, load weight and etc.

4.8 Communication Link

The communication link between the control center and the vehicle could be either a wired or wireless link. For wireless LAN (Local Area Network) connections IEEE

802.11 the standards exist for 2.4 GHz which in its latest iteration is 802.11n and the most recent 5 GHz technology is 802.11ac. The maximum throughput using 802.11n is 600 Mbit/s over three data channels and for 802.11ac the maximum is 1300 Mbit/s [33]. However when increasing the frequency used for transmitting data the range is shortened. This leads to that using 802.11ac with 5 GHz gives higher throughput but lower range [34]. There is more interference on the 2.4 GHz band since other wireless protocols use this frequency such as bluetooth, radio and microwave ovens. This will decrease throughput and range [35] together with an increasing number of packets lost when multiple devices are transmitting at the same time. Obstacles and interference with other devices have a direct impact on the range, therefore it is difficult to give a specific range for WLAN. A general rule [36][37] is that for 2.4 GHz the range is up to around 50 metres indoors and up to 100 metres outdoors and for 5 GHz it is approximately one third of these ranges.

4.8.1 Free-space Path Loss

The loss in signal strength of an electromagnetic wave can be expressed as Free-Space Path Loss (FSPL) and can be calculated in dB as

$$\mathrm{FSPL}(dB) = 20 \cdot \log_{10}(d) + 20 \cdot \log_{10}(f) + 20 \cdot \log_{10}\left(\frac{4\pi}{c}\right) \tag{4.1}$$

where d is the distance in metres, f is the signal frequency in Hz and c is the speed of light in m/s. So by keeping the FSPL constant, the distance can be calculated for some commonly used frequencies as can be viewed in Table 4.2. The FSPL is set constant to 70 dB and the frequencies used are 240 MHz, 2.4 GHz and 5 GHz, which is mid-range radio and Wi-Fi. As can be seen by using a lower transmission frequency the range can be extended. But with lower frequency the amount of data that can be transmitted is decreased. One way to utilize these properties is to send the heavy data transmission (camera images) over Wi-Fi and smaller but more critical commands (steering commands) over radio.

Distance (m)	**FSPL (dB)**	**Frequency (Hz)**
15	70	$5 \cdot 10^9$
31	70	$2.4 \cdot 10^9$
314	70	$240 \cdot 10^6$

Table 4.2: Free-space path loss for some frequencies at constant distance

4.8.2 Alternatives to Wireless

A wired connection will affect the maneuverability of the vehicle since the vehicle will only be able to follow one path and go back the same way in order not to tangle the cable. This type of communication is used in mines where trucks and diggers mainly follow the same path in a tunnel and the cable is managed on the vehicle as it drives. By using a wired connection a higher throughput and less latency can be achieved compared to a wireless link. The disadvantage in interference from other radio communication with wireless is reduced to a minimal since the data has its own medium to be transferred in with a cable. Network communication has overhead in the transmission which negatively impacts the latency. The overhead is considerably higher for wireless communication [38] due to more error checks and acknowledgements.

A combination of wired and wireless communication can be used. The main distance from the control center to the work site can be a wired link and the final distance at the site can be wireless to let the vehicle maneuver freely. If the wired part of a combined link is reasonably short, the whole connection link can be viewed as just the the wireless link. This is because the wireless link is slower and cannot carry the same amount of data as the wired one.

4.8.3 5th Generation Wireless Systems

Wireless communication systems are continuously developing and the fifth generation (5G) is the next major step. However, the systems will not be fully available until 2020 [39]. High expectations are set on this generation since more devices are connected with the advent of Internet of Things (IoT). Vehicle remote control is mentioned as a application of 5G. For safety critical systems such as vehicle communication, the intention is to reach latencies [39] as low as 1 ms and 10 ms in general.

A Pilot project called Pilot for Industrial Mobile Communication in Mining (PIMM) [40] consisting of a cooperation between Ericsson, ABB, Boliden, SICS Swedish ICT and Volvo Construction Equipment intends to implement communication using 5G to remotely control a Volvo truck for transporting ore in an underground mine started spring 2015 [41]. The program intends to initiate research that can be applied in a variety of applications and solutions within the usage of 5G.

4.8.4 Link Requirement

In this application the vehicle needs to be able to be maneuvered in all directions. A wired communication link will not satisfy this behaviour and therefore a wireless one is needed at the worksite. This will increase latency and decrease the amount of data that can be transmitted. The number of cameras used and other sensor data will set the requirements on how much data that needs to be transmitted from the vehicle to the control center.

Each of the used cameras (see section *5.3.1 - Cameras* for details) can transmit up to 16 384 Kb/s, and leads to four cameras transmits 65 536 Kb/s. By using half of that bitrate from the cameras, a total of 32 768 Kb/s, or ~32 Mbit/s which will be the minimum requirement for the communication. However, performing the stitching process (see *5.3.2.1 - Image Stitching*) onboard the vehicle and transferring the current view will reduce the amount of data needed to be transferred. The size of the stitched image presented to the operator will dictate the data needed. Lowering the requirement to 16 Mbit/s will account for a large viewing image and still lower the requirement by half. The data for controlling the vehicle (requested steering angle, speed etc.) will be significantly smaller. However, capacity requirements on the link depends on what sensor data that is transmitted back to the control centre. The most data consuming sensors following the cameras are the laser scanners (see section *5.3.5.1 - Light Detection And Ranging* for details). They transmit 720 floating points of 32 bits and sends these 20 times per second. This totals to ~0.5 Mbit/s. The odometry and GNSS data are another 20 data points which are also 32 bit floating points. That is negligible compared to the video and LiDAR data.

The round-trip-time for a byte of data using the communication link is set to a maximum of 20 ms. The transmission needs to be stable in terms of spikes in latency in order not to reach the threshold for lost connection which is specified to 200 ms. Violations of the thresholds for the communication link in terms of lost connection and packet loss has to be addressed. If the connection fails, the vehicle shall stop in order to avoid accidents of incorrect control signals.

4.9 Safety Requirements

Safety requirements are also needed to be specified. However autonomous construction vehicles will not have the same safety requirements as road vehicles since the work site will be a closed area. The speeds are often lower but safety and reliability still have to be considered. To minimize risks if the controls, sensors or communication fail in some way, a speed limit in the vehicle to not exceed a certain speed in both teleoperation and autonomous mode should exists. This speed limit is here arbitrarily set to 30 km/h. Furthermore an auto-brake system is required in both modes so that the truck will stop for obstacles. It should also be possible to override the emergency stop in teleoperation mode by coming to a full stop and disabling it. This is for instance if the LiDAR sensors are malfunctioning and making false detections. Emergency stop buttons inside the truck and in the control center are required.

5

System Design

By dividing the system into smaller subsystem the total solution can be scalable and flexible as parts can be added or removed as the system develops or requirements change. This full system consists of a vehicle and a control center for the operator, both described in the upcoming section. The used framework Robot Operating System (ROS) is described then followed by the subsystems described including the cameras with the stitching process, additional sensors and the autonomy in cooperation with the teleoperation. Lastly the simulation set-up using the simulation tool Gazebo is described.

5.1 System Architecture

The proposed system consists of two main parts. The user interface "Control Center" and the autonomous vehicle "The Truck". The user interface reads input from the operator and relays it to the vehicle. The vehicle returns sensor data and the stitched video stream to the user interface which are displayed in order to give the operator the best possible assessment of the vehicle state. The system is built up from smaller subsystems called *nodes* that communicate with each other. The main parts of the system can be seen in Figure 5.1 and are:

5.1.1 Vehicle

- **Autonomous** - The autonomous driver. Follows pre-recorded paths chosen by the operator and sent to the truck. Uses sensors to determine its location on the path and to avoid obstacles.
- **Cameras** - Four wide angle IP-cameras mounted on the vehicle with an overlapping field of view.
- **Camera stitching** - This node captures the streams from the cameras mounted on the vehicle and processes them in order to create one large image as described in section *5.3.2.1 - Image Stitching*. The operator can then pan the image in order to look around the vehicle.
- **Current path** - Stores the current path. It is used in two cases:
    1. **Autonomous mode** - A path that is to be followed is sent by the user interface from the **Path storage**. The autonomous node will then follow the loaded path.
    2. **Path recording** - When recording a path it is saved into the current path node. When the recording is finished, the path is sent back to the

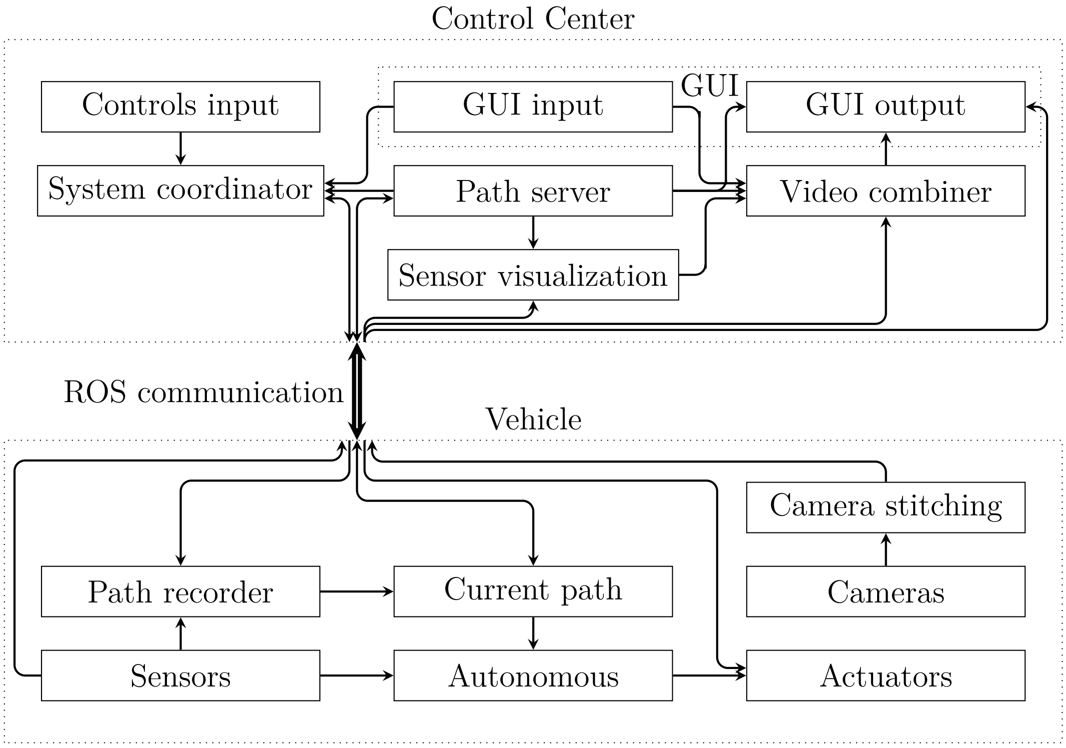


Figure 5.1: System overview of control center and vehicle with communication nodes

user interface and stored by the path server.

- **Path recorder** - Records when the vehicle is driven manually in order to be able to drive the same path autonomously when the driver commands it.
- **Sensors** - All the sensors on board the vehicle. This includes odometry, speedometry, RTK-GNSS, IMU, LiDAR. In addition to the sensor input some signal processing is done in this node, such as merging all LiDARs into one 360° scan.
- **Vehicle controls** - The actual controls of the vehicle. Steering, gearbox, handbrake, throttle, turn signals etc. These are controlled either by direct user input in the user interface or by the autonomous node.

5.1.2 User Interface

- **Controls input** - Reads input from different control surfaces such as a steering wheel or gamepads and translates the input to the appropriate data and passes it on to the **System coordinator**.
- **GUI** - The GUI is is used by the operator to interact with the vehicle in other ways than driving it.
  - **Output** - Autonomous status, position, mode, control and other information useful to the operator is shown here. A map with all available paths can also be shown. This can in the future be expanded with more information such as fuel level, running hours etc.
  - **Input** - The user can select options such as start path recording, choose

paths to drive autonomously, and select what is shown on the map and in the video.

- **Path server** - Stores all recorded paths available for autonomous driving and provides information to both **Sensor visualization** and **GUI** for presentation. Paths are sent to the **System coordinator** for autonomous drive and from the vehicle newly recorded paths are received to be stored.
- **Sensor visualization** - Images are created to visualize the sensor data in a human understandable way. For instance GNSS or other localization data is used to position the vehicle on a map, and LiDAR data is used to indicate obstacles. Paths from the **Path server** node are also drawn on the map to indicate where a autonomous operation can be initiated or completed.
- **System coordinator** - The node that dictates if the autonomous node or the operator is in control. It also handles the transition between autonomous and manual control.
- **Video combiner** - Combines the images created in the **Sensor visualization** node with the one from the **Camera stitching** node to create an augmented video feed.

5.2 Framework - Robot Operating System

All the different subsystems have to communicate with each other in a safe and reliable way with many different message types. This would be hard and time consuming to implement in an efficient way. The Robot Operating System[1] (ROS) is a open source framework for this that is gaining popularity and has done so during the past few years. It is a combination of communication, drivers, algorithms and other tools to aid creation of robots and vehicles. This leaves more time to the developers to develop new functionality and features, while safety and performance concerns are taken care of by the underlying system. Additional benefits are flexibility, scalability and ready made interfaces to other systems.

A typical ROS system is built up of many subsystems called *nodes* that send messages to each other. Nodes are easily added or removed depending on what the application demands. The nodes are written in either *C++* or *Python* and a vast library of existing nodes are available. However ROS is only a few years old, and has evolved significantly over the years the documentation available is often not complete and not always accurate.

5.3 Subsystems

This section describes the design choices and technical solutions of the subsystems of the whole system. Since the vehicle is operated out of sight, the operator needs to be able to track the vehicle in its surroundings. One way for the operator to assess the vehicle's placement is to use cameras mounted on the vehicle in order for the operator to see the surroundings. Another approach is to use maps where the

[1] `http://www.ros.org/about-ros/`

vehicle location is presented with surrounding areas, obstacles and walls. These two methods can be combined [42] to get a more accurate positioning of the vehicle. However, too many cameras, maps and other inputs for the operator may lead to loss of the surroundings [29] and reduce the performance. Studies have shown that using fewer screens but more accurate measurements gives better control of the vehicle[25][43]. The operator may suffer from tunnel vision when operating and concentrating on different screens simultaneously, which can lead to a loss of the surroundings instead.

5.3.1 Cameras

To create the surround view around the vehicle, four wide-angle IP-cameras will be mounted on the truck cab. They are placed so that the cameras overlap each other, so that the images can be combined to one large image. This is visualized in Figure 5.2. The cameras use an Ethernet connection to transmit the data stream over the Real Time Streaming Protocol (RTSP). This can then be fetched by a computer for processing. The cameras were part of the pre-existing hardware inventory and therefore used. The actual camera used can be viewed in Figure 5.3. The cameras can provide a resolution of either 1920×1080 or 1280×720 pixels in *H.*264 or *MJPEG* format. The bitrate can be chosen up to 16 384 Kb/s together with a maximum frame rate of 25 frames per second.

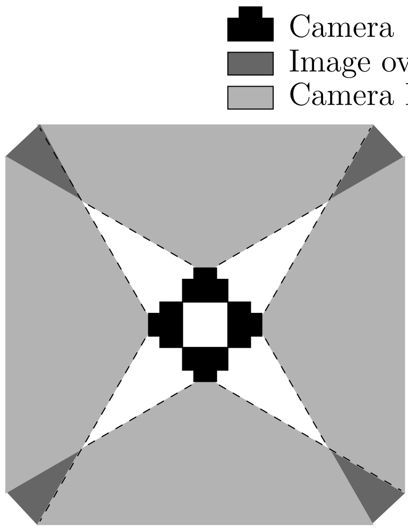


Figure 5.2: Four cameras with a 120° FOV. Mounted to capture a complete 360° view.

5.3.2 Image Processing

Image processing is done using OpenCV which is an open source library for image analysis and manipulation. It has support for Nvidia CUDA [44] for processing using the graphics processing unit (GPU). This is a major advantage when working with large amounts of data that has to be processes quickly such as images. The

Figure 5.3: IP camera used for surround view of the vehicle

GPU differs from the CPU in the way that it executes many calculations in parallel with thousands of simpler cores rather than a few powerful as in a CPU. OpenCV is also included in ROS (see *5.2 - Framework - Robot Operating System*) and can therefore be used directly in the simulation or it can be used standalone to process the streams.

The video streams from the IP cameras are processed and stitched together into one single stream with a 360° coverage. Information that is crucial to the operator is then overlaid on the stitched image. One proposed solution is to use a head mounted display (HMD) together with a spherical video feed. This can give the operator a "virtual cockpit" where it is possible to look around by moving the head. However this adds significantly more computations to the already demanding stitching process. The image must be warped to a spherical projection and displayed as two images, one for each eye. The head tracking has to processed and applied to the image. This will introduce more latency in the video feed and/or lower the frame rate [45]. Due to limitation of time and complexity a HMD will not be implemented. The solution that will be used is a setup with one or multiple monitors where the video stream can be displayed together with a graphical user interface (see *5.4.1 - Graphical User Interface*) with additional controls.

5.3.2.1 Image Stitching

A generic process to stitch images [46] is described below. Below this, the special case that is used in this implementation is described.

1. **Feature detection and classification** - The images are analyzed for distinct features and these features are saved for each image.

2. **Feature matching** - The features found in the images are compared to determine which images are overlapping and where.

3. **Image comparison** - Using the features found and matched in the previous steps, the homography matrices H for relating the overlapping images are calculated. H relates one image to an other so that the x and y coordinates for each pixel in the transposed image p'_x, p'_y relate to the original p_x, p_y according

to Equation 5.1.

$$\begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s_\alpha & h_x \\ s_\phi & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix}}_{H} \times \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \tag{5.1}$$

Where f translates the image, h scales it and s shears the image as can be seen in Figure 5.4

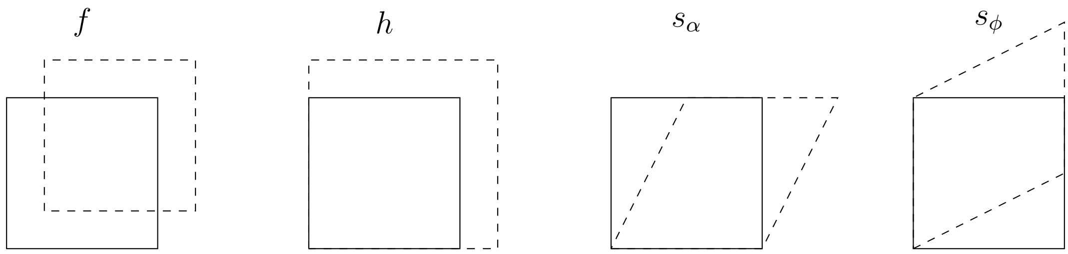


Figure 5.4: Illustration of the effects of the homography matrix.

4. **Image placement and transformation** - With the matrix H from above, overlapping images are transformed. They are then placed together so that features overlap each other.

5. **Blending** - To achieve a smooth transition between the images, blending is applied. A regular linear blend sets the destination pixel (D) to a weighted mean of the overlapping source pixels (S^1, S^2) as seen in Equation 5.2

$$D_{x,y,c} = S^1_{x,y,c} \cdot \alpha + S^2_{x,y,c} \cdot (1-\alpha)\ \alpha \in [0,1]\ \forall c, \text{ when } x, y \in \text{ blend area.} \tag{5.2}$$

where x and y are the position of the pixel and c is the color channel of the image. The blend area is dictated by the overlapping areas and the desired blend width. α varies from 0 to 1 in the desired area of the blend. A wider seam will smoothen out bigger subtleties such as exposure differences. If the images are not exactly lined up or the homography estimation is not perfect there will be *ghosting* in the seams of the images. Ghosting is when traces of a object can be seen a little transparent in multiple locations of the combined image.

One way to address this problem is to use multiband blending. The desired blend area is passed through a number of band pass filters. Then the different frequency ranges are blended separately in the same way as the linear blend. The high frequency part of the blend area will be blended with a short seam, and the low frequency area will be blended with a wider seam. This results in a less distinguishable blending.

6. **Projecting** - The produced image is an image laying flat in a 2D plane. This image can be projected using different mappings to suit the way the image will be displayed. For this application a cylindrical or spherical projection will be suitable to achive the feeling of looking around in the surrounding environment.

In this case the camera properties are known and their placement is static so steps 1,2,3 and 4 only has to be done once. The homography matrix can be saved and reused as long as the cameras do not move or are exchanged for cameras with other properties which reduce computation.

Performance Concerns

While manipulating an image in software the image is fully uncompressed and represented as a 3D matrix; $W \times H \times C$. W and H are the width and height of the image and C is the number of channels of the image. The number of channels of the image is called the color space and is usually three (Red, Green, Blue or Hue, Saturation, and Value) for color images and one for gray-scale images. Each element of the matrix represents the amount of each channel for each pixel. This is expressed either as a floating point number or an integer depending on quality and memory constraints. It is shown below that the amount of data that has to be processed quickly becomes large when image size and color depth increases.

As described in section *5.3.1 - Cameras* four cameras are used. These cameras can output images with the resolution of up to 1920×1080 pixels. The images from these cameras are represented with 3 channels of 32 bit floating point numbers (4 Bytes). Capturing the compressed images at 25 FPS and unpacking them into matrices in order for manipulation, the amount of data totals to around 2.5 GB/s (Eq 5.3).

$$W \cdot H \cdot C \cdot M_{\text{type}} \cdot n_{\text{cameras}} \cdot f = 1920 \cdot 1080 \cdot 3 \cdot 4 \cdot 4 \cdot 25 \approx 2.5 \ GB/s \tag{5.3}$$

Considering that the pixels then are to be manipulated, copied into one big image and blended, the amount of data that has to be processed quickly becomes multiple times the size of the initial captured images. Because the theoretical maximum throughput[2] of used computers (DDR3 memory) is 12.8 *GB/s* it is apparent that the computer's performance can become a bottleneck, especially if it is doing other computations parallel to the stitching.

5.3.2.2 Information Overlay

When the operator is driving the vehicle the primary view is the stitched video stream. Information that is important to the operator will then be overlaid onto the video so it can be seen without looking away from the video stream. A map is shown in the top right corner. In the lower left corner information about and distance to the current chosen path is presented and in the lower right corner a speedometer is displayed. This can be seen in Figure 5.5. The overlays are semi-transparent so it is be possible to see objects behind. The process of blending an image onto another is done by calculating a weighted average of the two overlapping pixels from the two source images. The weight is called a mask and is a grey scale image. By performing a threshold operation on the image to be overlaid the mask is created only where there is image information. This part is set to a grey value allowing information

[2] `http://www.crucial.com/usa/en/support-memory-speeds-compatability`

from both images to be visible. The operator can customize the overlays and choose what is shown.

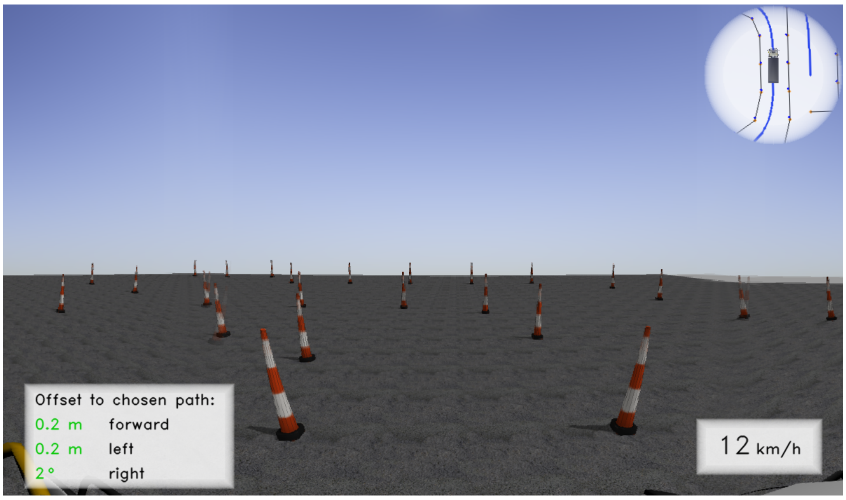
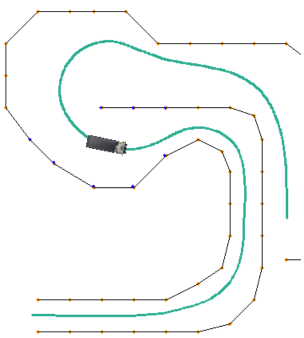


Figure 5.5: 110° Stitched camera view from vehicle in simulation with map and offset to chosen path.

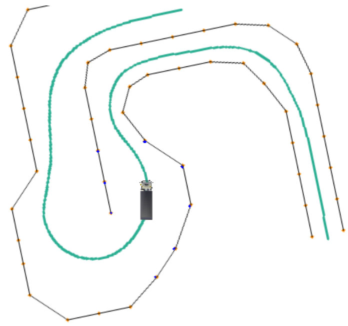
5.3.3 Maps

Using a map where the operator can view the vehicle from a top-down perspective, gives the operator an overview of the area to simplify navigation. The alignment of the map can be either fixed or rotating. If the map is fixed and the vehicle rotates, humans tend to rotate the map in their minds [47] in order to position themselves. Using a rotating map instead, where the vehicle is fixed with the front pointing upwards has been proven [48] to be better for remote control and maneuvering. The map can either be produced beforehand or be created as the vehicle travels. A predefined map will be more accurate but if the surroundings are changing over time there is a benefit of creating the maps while moving. One of the more popular methods for creating these maps is SLAM [49] where the vehicle is able to both create and at the same time keep track of itself in the map.

Because the area where the vehicle is going to operate is known, the map is created beforehand. Then it is used as a background with the vehicle inserted into it. Because of the high accuracy of the positioning system and the pre-produced map the vehicle's position is presented very exact. Creation of the map together with the vehicle and information data is done in OpenCV. Two maps are created in the same node with one map fixed with the vehicle itself moving in it. The other map rotates around the vehicle which is fixed pointing upwards. The different maps can be viewed in Figure 5.6. This gives the operator the choice of change between these two maps during operation, and the different maps can be shown in different environments such as the GUI or overlaid in the video. In addition to the vehicle itself the LiDAR sensor data is drawn in the map and in Figure 5.7 it can be seen how the sensors scan the environment in the simulation and how it is presented to the operator. The LiDAR data provides useful information on how accurate the



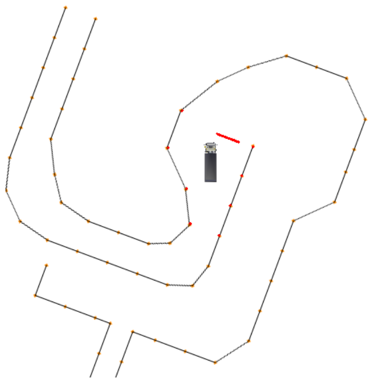
(a) Fixed map with rotating vehicle with north upwards.



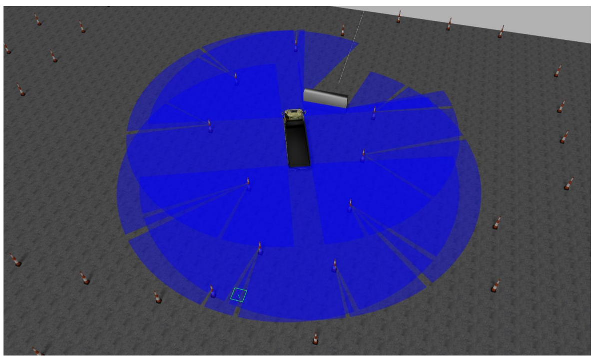
(b) Rotating map with fixed vehicle pointing upwards.

Figure 5.6: Overview maps with surroundings and recorded paths.

positioning of the vehicle in the map is. But the primary purpose is so that obstacles that are not in the map are drawn. This could be other vehicles or other objects. Depending of the distance the color changes from green at a safe distance via yellow to red if it is dangerously close. The stored paths are also drawn out on the map. This is both to aid planning the use of autonomous functions, and to help navigate to a selected path.



(a) Map with obstacle detection.



(b) Obstacle and laser scan from simulation.

Figure 5.7: LiDAR sensor data presentation.

5.3.4 Control Inputs

Different types of control inputs are implemented in the system to have the ability of evaluate the performance implication from the different controls. A interface for a normal steering wheel with pedals for throttle and brake made for computer games is implemented. Further, two different gamepad controllers are interfaced alongside with a traditional computer keyboard. In addition to the controls for steering, acceleration and braking, commands for zooming in the map in the video stream are implemented. When presenting the video stream on a screen that will not fit the full 360° view, controls for pan in the image are available on the controller.

5.3.4.1 Haptic Feedback

When the operator is separated from the actual vehicle the controllers lack the direct coupling and feedback is therefore missing. One way to address this is to implement haptic feedback in the controllers to simulate the coupling. Feedback will enhance the operator's performance [50] since the somatic senses can be used instead of just the vision on the screens. To achieve accurate haptic feedback, low latency is important for stability and performance together with correct modelling and controlling of the system. It may not even be possible to achieve if latencies are too large. However, due to latency concerns, haptic feedback is not implemented in the suggested system.

5.3.5 Additional Sensors

As well as the cameras mounted on the vehicle there are a number of additional sensors present for the autonomous functions. These are LiDAR sensors for range detection, GNSS sensors for position and heading and an IMU for measuring acceleration and angular rate. These sensors are also used for teleoperation and are described below.

5.3.5.1 Light Detection And Ranging

Light Detection And Ranging (LiDAR) sensors are laser scanners which scan the surroundings by emitting a short but intense laser pulse and measuring the time for it to return. From this time it can calculate the distance to the reflection. This laser beam is rotated to create a 2D image of the surroundings. Four SICK LMS111[3] LiDAR sensors are mounted in each corner of the truck. Each of them has a detection sector of 270° and a range up to 20 metres with a resolution of 30 mm. These scans are composed to one single 360° scan of the surroundings on the vehicle before transmitted to the control center with a total number of 720 scan points. The LiDAR data is used for positioning, emergency braking and obstacle avoidance by the system, and presented to the operator to aid navigation. A study [51] of teleoperation with a small robotic vehicle shows that the users were able to navigate more accurately and quickly using solely sensors displaying the surrounding environment instead of a fixed mounted camera on a robot. The LiDAR data will be displayed on the map, see *5.3.3 - Maps*

5.3.5.2 Global Navigation Satellite System

Global Navigation Satellite System (GNSS) is an umbrella term for technologies for positioning via satellite including GPS, GLONASS etc. It works by multiple satellites with known locations in orbit emitting the current time. The receiver then

[3] `https://www.sick.com/media/dox/5/15/415/Product_information_LMS1xx_2D_laser_scanners_en_IM0026415.PDF`

reads this time and compares it to its clock and using this offset calculates how far away each satellite is. With this knowledge about multiple satellites the receiver can calculate its position. The more satellites that the receiver can see, the more exact is the calculated position. This is used to track the vehicle's position in the world frame in order to navigate and visualize this information on a map. The system consists of a primary GNSS unit and a secondary antenna. With this setup both position and direction can be measured. The system has support for Real Time Kinematic (RTK) GNSS and network RTK. This is a system [52] that measures the phase of the GNSS carrier wave instead of the actual data. This wave is then compared to the phase at a know location. This technology allows positioning with a few centimeters accuracy compared to meters with conventional GNSS. A major limitation of the technology is that it only works close to the reference point. If there is a network of known reference points with GNSS receivers over a large area the phase of the carrier wave at a specific location can be calculated and set to the receiver. This is known as network RTK and can be used if the vehicle is to be used in large areas, or different areas where there is not a possibility to install a new reference point.

5.3.5.3 Inertial Measurement Unit

An Inertial Measurement Unit (IMU) is used to measure the orientation of the vehicle in three dimensions using accelerometers, gyroscope and magnetometer. The accelerometer measures change in velocity, the gyroscope measures the change in angles (roll, pitch and yaw) and the magnetometer is an electronic compass measuring the earths magnetic field. Using these measurements the attitude of the vehicle can be accessed. During tests of a teleoperated vehicle performed by the US Navy the most common incident was almost-roll-over accidents [45] where lack of attitude perception was the biggest contribution to the incidents [53]. It has been shown [54] that an operator tends to navigate more efficiently using a camera view that moves with respect to the vehicle but stays fixed perpendicular with gravity. This is compared to a camera fixed to the vehicle with a roll attitude indicator overlaid the video feed. Because of the used fixed camera configuration if a dangerous angle is read, such as driving with a large sideways tilt, this will be displayed to the operator.

5.4 System Coordinator

The nodes in the system are coordinated by a coordinating node. It keeps track of the states of the system and issues commands depending on the inputs it receives. The main interaction with the operator is through the GUI.

5.4.1 Graphical User Interface

The Graphical User Interface (GUI) is built with the open source tools Qt[4] and PyQt[5] and can be seen in Figure 5.8. The visually largest part is the map, it is similar to the one shown overlaid in the video feed, but larger. Below are controls for the maps, where both the overlaid and GUI maps are controlled. The operator can there choose between rotating or fixed map and if LiDAR data and/or paths are to be drawn on the different maps. Zoom controls are also available for the GUI map.

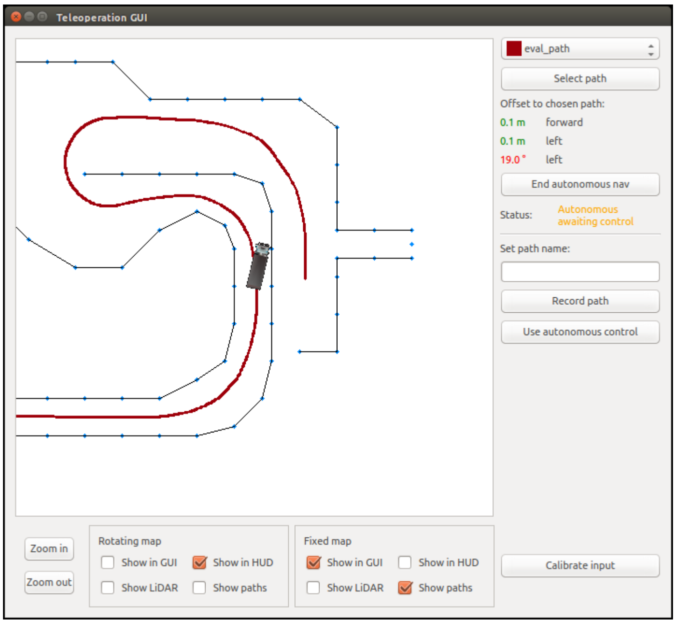


Figure 5.8: Screenshot of the GUI

To the right of the map are the controls for the autonomous functions. At the top is a drop down menu where a path can be chosen from all available paths. In addition to their name, their color correspond to the color of the path drawn on the map. When a path is chosen, autonomous navigation can be started by pressing *Start autonomous nav*. Information on how to align to the path in order to start navigation is shown in both the GUI and in the video feed (see Section *5.4.2.3 - Paths*). This button is also used to abort autonomous navigation and reads *End autonomous nav* when navigating autonomously. When properly aligned to the path, the operator can hand over to autonomous control with the button *Use autonomous control*. This button is also used to manually take over control. To record a new path in manual mode a name of the path is given and the button *Record path* is pressed. After driving, pressing *Stop recording* will store the path for later use.

5.4.2 Autonomous Functions

The truck is able to follow pre-recorded paths. The paths are created by record-

[4]https://www.qt.io/qt-for-application-development/
[5]https://riverbankcomputing.com/software/pyqt/intro/

ing while driving the vehicle. Below the functionality is described together with synchronization between autonomous and teleoperation mode.

5.4.2.1 Navigation

When navigating in autonomous mode the vehicle follows pre-recorded paths. Using the on-board sensors it scans the surroundings to estimate its position in the pre-made map. If satellite positioning is available this is used as well. Available paths are displayed in the map, and to start autonomous navigation of these paths is chosen in GUI. The truck then needs to be driven manually to the beginning of the path. Distance and angle offset to guide the driver to the correct position and alignment is presented to the operator as head-up information in the video feed. When the truck is positioned correctly the autonomous navigation can be initiated. While driving, if the truck senses an obstacle or faces other problems it will stop and wait for the obstacle to disappear or for an operator to start manual control. When driving manually, autonomous navigation can be resumed by the operator stopping the vehicle on the current path and switching over to autonomous mode again.

5.4.2.2 Synchronization

To prevent dangerous behaviour from the vehicle when switching between control modes, some simple rules for the implementation has been set and are here presented. Switching from manual teleoperated control to autonomous drive can only be done when the vehicle is stopped on and aligned to the chosen path. Then autonomous mode can be initiated. When the autonomous driver has confirmed that the position is valid and that navigation from there is possible, control will be granted to the autonomous functions. When a request for manual control is sent to the vehicle it will stop before handing over the controls. This can be overridden if the truck is on its way to collide with something it cant see or that the autonomous functions are failing in some other way. If the navigation is interrupted by manual control autonomous navigation can only be resumed if the vehicle is stopped on the current path. When the truck has reached the end of the path used for navigation, it will stop and wait for the operator to take further actions.

5.4.2.3 Paths

When a path is chosen, the operator needs to drive to a point of that path in order to initiate autonomous functions. In addition to the map the parallel and perpendicular distance offset to the closest point on the path is calculated and presented. The angular offset between the vehicle's current heading and the heading required by the path is also displayed. The closest point is calculated as a straight line regardless of walls and obstacles. This is intended to be used in addition to the map for a more precise positioning of the vehicle. Presentation of this information can be seen in Figure 5.5 and 5.8. The information is red until the vehicle is inside the set threshold that is needed to initiate autonomous navigation, then it is set to green. Only when all three parts; parallel, perpendicular and angular offset are inside the threshold autonomous navigation can be initiated.

The paths used for navigation are needed in several subsystems. Naturally the autonomous path-follower needs the path to use them for navigation. But they also needs to be drawn into the map, presented in the GUI and they are needed to provide navigation assistance to the operator to drive to the path. When a path is recorded it is stored as plain text files consisting of the points or "bread-crumbs" where each of the points includes position together with the vehicle angle and the speed at that particular point. Instead of all nodes that need this information knowing the location of the files and have to access the file system a path server loads all paths. Nodes that need the paths can then request the information that is needed.

5.5 Gazebo Simulation

Bundled with ROS is a simulation tool called Gazebo. Gazebo includes physics engines, high quality graphics and integration with ROS. This makes it straightforward to test the system built for the real vehicle directly with the simulation without major modifications or additional software.

A complete model of the vehicle with control interfaces and existing sensors is set up to test and evaluate the features of the system before moving to a real vehicle. The model is implemented to simulate the Volvo truck described in section *2.2 - Evaluation vehicle* with the same dimensions and properties. A screenshot of model in Gazebo can be viewed in Figure 5.9.

The physics engine in Gazebo does all the calculations, so the major work in building the simulation is defining the model of the vehicle and the world. A model is built using building blocks called *links*. These can have different properties, but the most basic are *visual, inertial* and *collision* and more about this can be seen in *5.5.1 - Visual* , *5.5.2 - Mass and Inertia* and *5.5.3 - Collision*. These links are then fastened together with what are called *joints*. The joints can be of different types depending on how the links should interact with each other which is elaborated on in *5.5.4 - Propulsion*. The world is built in a similar fashion, but with multiple models pre-defined in Gazebo, see *5.5.7 - World*. When the model and world is built and added, the inputs to the simulator are throttle, brake and steering. The simulator outputs a visual 3D view of the vehicle in the world, poses for all links, and the outputs from all sensors.

5.5.1 Visual

The basic building blocks when building a model are called links. A link in Gazebo can be defined by either basic shapes or what is called *meshes*. These meshes are created in Computer-Aided Design (CAD) software and exported to a shape built up from many small polygons. This model is created from a CAD drawing of the real truck, divided into three parts. The truck, the load bed and a wheel. The wheel is then added eight times in different poses. For performance reasons all parts have been greatly simplified to be drawn by only around 5% of the original polygons creating the mesh. The visual part is used for the visual representation in Gazebo, LiDAR reflections and what the modelled cameras can see. The visual part of the truck can be seen in Figure 5.9.

Figure 5.9: The model of the Volvo FMX truck in Gazebo simulation.

5.5.2 Mass and Inertia

The mass and inertial model is made simple for both performance concerns and because it a very exact model is not required in this application. The real truck weighs around 22 000 kg [55] and this weight has been distributed in three blocks and four axles and eight wheels as can be seen in Figure 5.10. The wheels weigh 50 kg and axles 150 kg each. The chassis has been modeled to weigh 4 000 kg, the cab and engine 6 000 and the bed 10 000 kg.

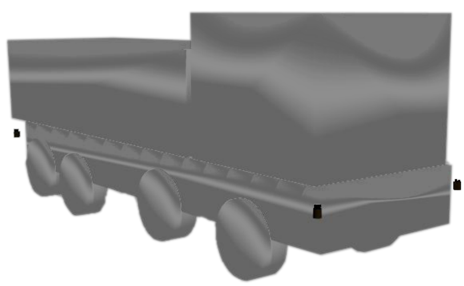


Figure 5.10: The inertial model of the truck.

5.5.3 Collision

The collision model dictates when the model is touching another physical object in the simulation. As in the previous sections, for performance concerns the collision model is a greatly simplified model of the truck. The collision model is created as a few simple shapes created in a CAD software and exported as a mesh with very few polygons. The collision model can be seen in Figure 5.11.

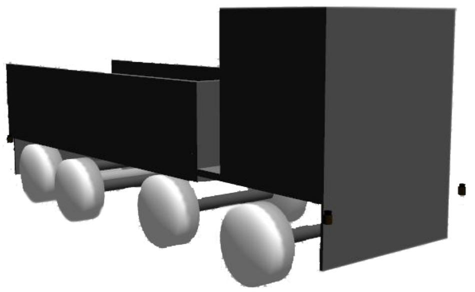


Figure 5.11: The collision model of the truck.

5.5.4 Propulsion

The real FMX truck can raise its second and forth pair of wheels when they are not needed to improve maneuverability and decrease fuel consumption. The truck is modeled with these pairs raised, hence it has four wheel drive. As seen above the model is built from links and joints. The joints connecting the links together can be of different types. The most common is a fixed joint which is a rigid connection between the links. The wheels are connected with the axles with a joint called *continuous*. It is a joint that can rotate continuously around an specified axis. The joint can be specified to have an maximum angular velocity and a maximum torque. The angular velocity set to represent 30 km/h linear movement of the truck as is specified in *4 - System Requirements*. The maximum torque is set to 2400 Nm which is the maximum torque of the Volvo FMX D13 engine. Connected to the joint is a simple PID controller and the desired value to the controller is controlled by the throttle. There is no gearbox modeled, since the gearbox in the real truck is a fully automatic gearbox, and such realism is not needed from the model.

5.5.5 Steering

The wheels used for steering are connected to the truck with joints called *revolute* which are hinge joints that can be specified to have a certain range of motion around an axis. A position PID controller is connected to the joint setting the steering angle of each wheel. The steering is implemented using an ackermann steering model which is illustrated in Figure 5.12. The angles for each wheel is calculated by the following equations:

$$R = \frac{L}{sin(\delta)} \tag{5.4}$$

$$\delta_{in} = tan^{-1}\left(\frac{L}{R - \frac{D}{2}}\right), \qquad \delta_{out} = tan^{-1}\left(\frac{L}{R + \frac{D}{2}}\right) \tag{5.5}$$

where L is the wheelbase, D is the axle width, R is the turning radius and δ the desired steering angle. δ_{in} and δ_{out} are the actual wheel angles for the inner and outer wheel. For $L << R$ Equation 5.5 can can be simplified as $\delta_{in,out} \approx \frac{L}{R \pm \frac{D}{2}}$. The maximum value of δ is 30° to represent the same maximum steering as the real truck. The controllers are tuned to be very responsive to the desired angle of the wheel. The dynamics of the steering is then modeled together with the calculations of the Ackermann angles.

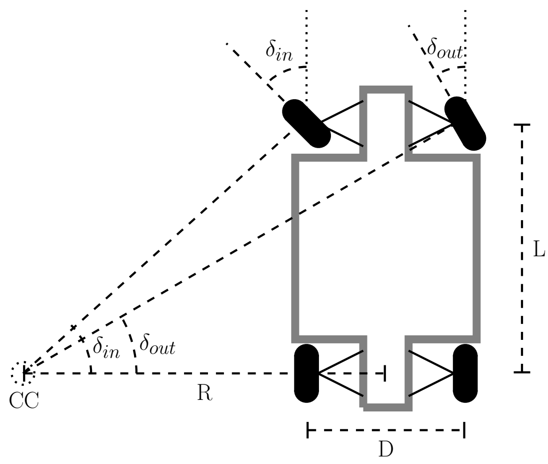


Figure 5.12: Illustration of ackermann steering, *CC: center of turning circle.*

5.5.6 Sensors

Four LiDAR sensors are mounted on the truck, one in each corner as described in section *5.3.5.1 - Light Detection And Ranging* to get a 360° view of the surroundings. The modelled version uses a pre-existing ROS package and is set to have same properties as the actual lidars used.

The four cameras are mounted on the cab to get a full 360° view. The placement of the cameras has been varied to test the best positioning, both in regards to cover as much of the surroundings of the vehicle as in to give the operator a good sense of position. The camera models in Gazebo are simple and cannot fully model the used cameras. The basic properties are modeled, such as the resolution and frame rate, 1920x1080 at 25 frames per second. But the warped image produced from the fish-eyed lenses is difficult to recreate, and a wide but straight image is emitted. To achieve this warped image as in the real cameras the video feed is processed in OpenCV. This produces a more realistic video feed at the expense of image quality.

There exists no modelled RTK-GNSS in the simulation, however the position of the truck can be measured directly in the Gazebo simulation. This is used as GNSS data even though noise is not modeled. The Gazebo world uses a different coordinate system than GNSS where everything is measured in meters and a origin in the middle of the world used. A node that translates the GNSS data to meters with an origin that can be set will has to be used for tests in real world.

5.5.7 World

The world used to test the system is a large asphalt field where a track has been set up using a number of cones. The layout can be viewed in Figure 5.13 and is designed to represent tunnels or partly narrow roads. The cones are tall enough for

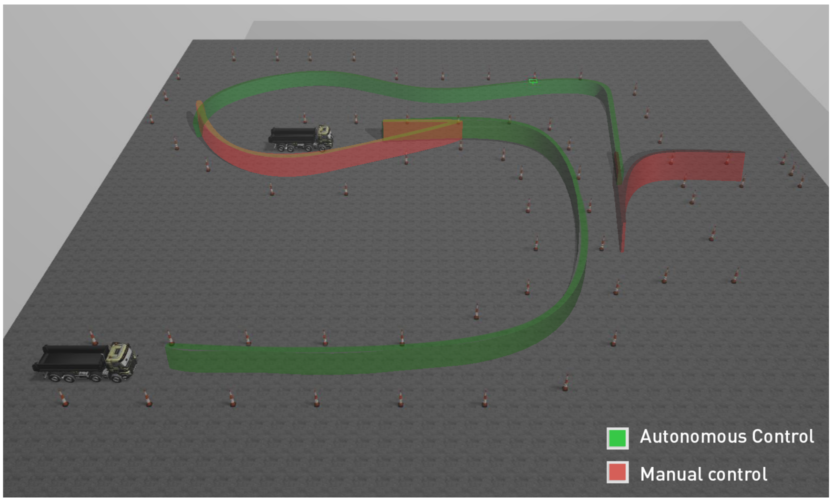


Figure 5.13: Layout of the track used for testing

the LiDAR sensors to recognize them. The course starts at the bottom left with two curves where the truck will drive autonomously until it reaches an obstacle in its path. At this point the track is a bit wider. It will then stop and the operator will have to take over and drive around the obstacle manually. After the obstacle the operator drives the truck back on the path to resume autonomous navigation. This is supposed to simulate another truck at a meeting point and will test interruption and resume of the autonomous functions together will manual control. The truck will then drive autonomously two more turns until it reaches the end of the path. The operator will then resume in manual control and reverse into a small space. This is to test how much the surround vision and sensor data supports the operator, after this maneuver the track is complete.

5.5.8 Performance

The model has been compared to data from a real FMX and behaves as expected. Collisions work as expected when diving into obstacles. The truck accelerates to 30 km/h in about 6 seconds. A real FMX does this in about 5 - 8 seconds depending on engine and load. The turning radius is 11 meters which is on par with the real truck [55].

6
Results

It has been found that in this application the most crucial functionality for teleoperation is when an obstacle is in the path for the autonomous truck. The teleoperation functionality has been used to define new autonomous functions for repetitive tasks which has been proven to work well. This could be a task which is such that one part is simple and repetitive, and one part is more challenging, for instance driving a long road, and then empty the load in varying places. It has been shown to be effective to define an autonomous task for the driving, and when the truck is at the unloading place where the vehicle is not capable to do the unloading autonomously an operator can handle it via teleoperation.

Teleoperation can also be used to recover an autonomous vehicle that has either been damaged or lost track of its position in the map. However, if sensors are damaged it can be harder to assess the state of the vehicle and determine if it is safe to operate without causing more damage to the vehicle or the surroundings. If the truck has lost its position in the map, it can be more difficult for the operator to drive it since the aid of the map will be lost.

When using teleoperation the direct coupling to the controls is missing and the somatic senses can not be used while driving. Many industrial vehicles today have a mechanic connection to control steering and pedals. Haptic feedback could be implemented to assess this problem. New machines and vehicles coming out to the market have started to use steer-by-wire systems where the controls are sensors that have artificial feedback from electric motors. Using this same feedback in a teleoperation setting could solve this disconnection, though latency can be a problem.

6.1 Standards and System Layout

There are several standards associated with teleoperation and remote controlled vehicles such as *ASTM E2853-12* [56] which defines a test method to evaluate how well a teleoperated robot can navigate in a maze, or *ISO 15817* [57] which defines safety requirements for OEM remote controlled earth moving machinery. Neither of these nor any other standard found apply to this prototype. Standards regarding communication or how autonomous industrial vehicles communicate could not be found. What was found is that when not using proprietary solutions the Robot Operating System (ROS) is the most popular solution in the industry when creating autonomous vehicles.

Building the system in a modular fashion with a node for every function makes it simple to exchange only the part that are specific to a certain vehicle or application. For instance a node calculating a certain yaw angle of the vehicle to a steering wheel angle can easily be exchanged to a node calculating the two inputs to a set of tracks on an excavator. This modular architecture also makes it easier improve the system by upgrading single nodes and makes it more stable since if one node crashes, it can be restarted while the rest of the system keeps running. However, building the system with many nodes can add significant performance overhead to the system since the nodes have to be synchronized and communicate with each other with different messages types.

6.2 Autonomous synchronization

The vehicle has two modes of control, manual and autonomous and can be set in a stopped mode. These states together with their transitions can be seen Figure 6.1. Initially the vehicle is in stopped mode and from there autonomous (transition *a*) or manual control (transition *b*) can be set. When in manual control the operator has full control over the vehicle. The auto-brake system from the autonomous driving system is still active, so if the operator is on its way to collide with something the vehicle will stop. This can be overridden if for instance a LiDAR sensor is broken and giving false readings that makes the truck stop. These states are Manual Safe, and Manual Unsafe in Figure 6.1, with the transitions *h* and *i*. Similarly there is Autonomous Safe which is autonomous control with auto-brake for obstacles and Autonomous Unsafe that does not brake automatically. This state is never used and therefore forbidden. When stopped in the manual modes, stop mode can be entered via transition *c* or *j*. If the truck is not stopped when requesting stop mode manual mode will be entered via transition *e* or *i*. To start autonomous navigation

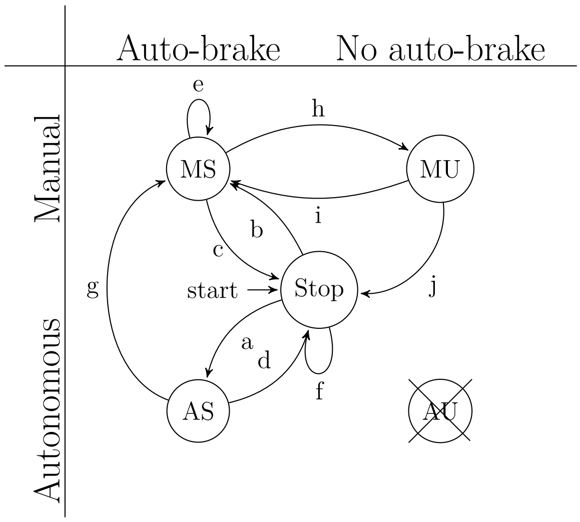


Figure 6.1: A state diagram showing the modes of control and autonomous synchronization. The states are **M**anual **S**afe, **M**anual **U**nsafe, **A**utonomous **S**afe, **A**utonomous **U**nsafe and Stop. The transitions are described in *6.2 - Autonomous synchronization*

a path is chosen from the existing pre-recorded ones. The truck needs to be driven to the start of the path and stopped in order to go to Autonomous Safe state via transition *a*. If the truck is not aligned correctly it cannot enter autonomous mode and it will stay in stopped mode via transition *f*.

The system can send a request for manual control while driving autonomously. The vehicle will then come to a stop and then switch to manual control for a safe hand over, this is transition *d* and *b*. This can be overridden if the the operator notices an emergency and has to take control immediately to prevent an accident as seen in transition *g*. To resume the autonomous navigation after manual control the truck is driven onto the current path and stopped again (transition *c*), and a request can be sent to the system to regain autonomous control (transition *a*). Similarly as when starting an autonomous task the truck has to be aligned correctly. When the autonomous task has ended the vehicle will stop, transition (*d*) to stopped mode and wait for new commands.

6.3 Evaluation

The evaluation is performed inside the simulation environment described in *5.5 - Gazebo Simulation* using the predefined course. Different support functions, two types of controls, the impact of varying amounts of latency and frame rates are tested by letting a number of test subjects drive the course. They where timed, their behaviour was observed and afterwards they where interviewed. The system requirements specified in chapter *4 - System Requirements* are verified to assure that the system and simulation is suitable for this evaluation. The results can be seen in Table 6.1.

As can be seen most of the requirements are fulfilled apart from that no gearbox control nor handbrake is implemented in the simulation model. Also, indication of the attitude of the vehicle has not been implemented due to that the tests are performed on a flat surface.

6.3.1 Driving Experience and Support Functions

Running the simulation using the predefined cone-track with all the supporting functions switched on has shown to work well. The natural choice is to use the stitched camera image most of the time while driving the vehicle. But when driving through narrow corners and close to obstacles the support of maps and proximity sensors helps to inform about the surroundings for precision driving. Turning off the support functions and only using the camera feedback works but causes the operator to slow down slightly in order to pan around in the 360° video to get an overview of the vehicle placement.

Generally, users kept the video feed set straight forward and only panned around when reversing or if in a tight passage when the map with the distance indication was missing. The benefits of the stitched 360° video feed compared to a number of fixed cameras in strategic places that can be toggled between is not obvious. The

Table 6.1: Requirements on the system for teleoperation, priority, verification with results

Criteria	Value	Variable	Priority	Verification	Fulfilled
Autonomous synchronization					
Manual takeover from autonomous			1	**S**	Yes
Resume autonomous after manual			1	**S**	Yes
Autonomous start	From path start		1	**S**	Yes
	Anywhere on path		2	**S**	No
Autonomous tasks					
Record new paths					
in teleoperation mode			2	**S**	Yes
Communication link					
Latency	Max 20 ms	Yes	1	**L**	Yes
Data capacity	Min 17 Mbit/s	Yes	1	**C**	Yes
Orientation Map					
Fixed map and rotating vehicle	On/off		2	**S**	Yes
Rotating map and fixed vehicle	On/off		2	**S**	Yes
Representation of LiDAR data	On/off		2	**S**	Yes
Sensor data presentation					
Speedometer	Visible		1	**S**	Yes
Vehicle attitude	Visible at danger		3	**S**	No
Distance to obstacles	Visible when close		2	**S**	Yes
Proximity warning	Visible when close	On/off	3	**S**	No
Teleoperation					
Speed limit	30 km/h		1	**S** & **T**	Yes
Desired steering angle			1	**S**	Yes
Desired acceleration			1	**S**	Yes
Desired breaking			1	**S**	Yes
Gear box control			2	**S**	No
Parking brake control			2	**S**	No
Control types	Steering wheel,		1	**S**	Yes
	Gamepad, Joystick		2	**S**	Yes, No
Video					
Latency	max 500 ms	Yes	1	**I**	Yes
Frame rate	min 15 FPS	Yes	1	**I**	Yes
Field of view	360°		1	**S**	Yes
Image quality	Road sign, 15 metres	Yes	2	**T**	Yes

T = Live test, **S** = Verify in simulation, **I** = Implement meter, **L** = Measure with *ping*, **C** = Measure with *iperf*

advantage of this technology is probably much greater if combined with a HMD to create a more virtual reality like cockpit.

When using only map when driving, the operator tends to lower the speed driving around the course. The rotating map appears to be more convenient since the steering inputs will always be the same when turning. In the fixed map with the vehicle rotating the operator rotates the map in the mind and sometimes left becomes right and vice versa. This result has also been found in earlier studies [48]. When using the fixed map, cutting corners were more frequent causing more cones being hit than using the rotating map, even at lower speeds.

The actual size of the vehicle was difficult to perceive using only the cameras, especially the width of the vehicle driving in narrow roads. Since the cameras are mounted on the cab, the actual truck is not visible when driving. Therefore two lines are introduced in the front camera image to point out the width of the vehicle. If obstacles are outside of these lines, there will be no impacts.

Using a gamepad for control input the test results depends on the experience of the driver. A driver that has played a lot of video games can skillfully control the truck with the gamepad while an inexperienced driver tends to often use the joystick inputs at full strength. Using a steering wheel, drivers tended to use the controls in a more conservative manner leading to more precise control. Also using the wheel drivers did not expect the steering to react immediately as was the case with the game pad. This is believed to be because moving the joystick from full left to full right only takes a split second, as with the steering wheel it takes around a second, and the actual wheels of the truck more than so.

Findings have come to that the driver adopts to different scenarios and after some practice the different support features tends to be less of use. The driving speed is also increased after a few laps around the course since the driver gets used to the controls and starts to learn the course. For driving longer distances the camera view is beneficial over just using the map since speed is higher. However just maneuvering around an obstacle to then continue autonomous driving, a map with range detection is sufficient to handle the task. Since the LiDAR sensors only measure in a 2D plane and has a range of 20 metres, relying only on the predefined maps and sensors can be dangerous. Small objects that does not reach up to the sensors can not be seen, for instance a fallen cone. Driving in a tunnel where the walls are not smooth, the LiDAR sensors may detect an indentation and therefore sense that the tunnel is wider than it actually is. Therefore the usage of several different sensors and support functions and letting the operator interpret and combine these are safer.

6.3.2 Impact of Latency

Tests show that for latencies smaller than around 300 ms the drivers can compensate for the latency and there is not much change in efficiency and control. As can be seen in Figure 6.2 the effect is 18 % increase of completion time around the course when introducing 250 ms latency. As the latency reaches above 300 ms, drivers tend to control by issuing an input and then waiting for the effect until the next input is issued, known as stop-and-wait behaviour. This can be seen as a jump in Figure 6.2 between 250 and 500 ms. With 500 ms latency the completion time increased with 47 % and with 58 % at 750 ms latency. The degree of the stop-and-wait control increases with the amount of latency as well. During the tests the vehicle was controllable up to 1000 ms in delay, with higher latency nobody could complete the course. It was noticeable that driving in constant curvature corners was easier than in narrow straights since it was difficult to keep the vehicle driving in a straight line. The driving tended to be "snake-like" and the amplitude of the oscillations increased with latency since the driver tends to overcompensate steering input.

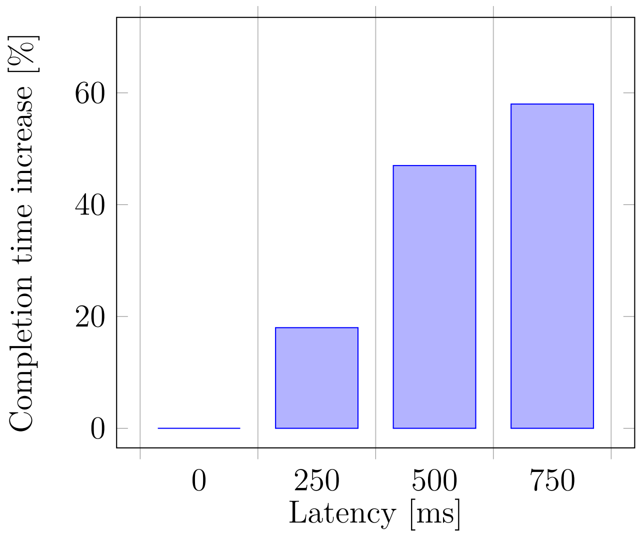


Figure 6.2: Completion time increase in percent due to higher latency.

When latencies increased the driver tended to drive more slowly through the course since the difficulty increased. This lead to that the there were few violations when the latency increased until a point where it got undrivable above 1000 ms. One of the more surprising discoveries was that when the latency increased the speedometer would aid the driving since the perception of the current speed was lost when latency was introduced. By using the knowledge of the speed the right amount of steering and throttle/brake input could be applied to the vehicle to complete the course.

Because the stitched image is created using the computer in the vehicle and only the part of the image the operator looks at is sent back, the latency affects the controls of this as well. This made it very hard to pan precisely in the image, and a majority of the test subjects found it harder to control the camera angle then to control the vehicle at large latencies.

The cameras capture images at 25 frames per second and by lowering the frame rate, the tests have shown that the controllability of the vehicle does not decrease with frame rate as long as the frame rate stays over 10 FPS. However during tests with low frame rate drivers report getting more mentally exhausted and need to focus more to achieve the same results as with a higher frame rate. The distance the vehicle travels between two frames for a acceptable frame rates is significantly lower then the distance traveled before the user can observe it due to acceptable latency. This can be seen in Table 6.2, for instance when the speed is constant at 30 km/h and the frame rate is as low as 10 FPS the distance is reasonably small (below one meter) compared to a small latency of 250 ms where the truck is 2.08 meters ahead of the video stream.

The tests subjects preferred driving with lower frame rate compared to larger latencies. Due to that the communication link cannot transfer the required amount of data, lowering the frame rate could be one way to keep latency low and consequently driveability higher.

Table 6.2: Traveling distance between video frames and at different latencies at 30 km/h.

FPS	Distance [m]
25	0.33
20	0.42
15	0.55
10	0.83
5	1.16

Latency [ms]	Distance [m]
250	2.08
500	4.17
750	6.23
1000	8.33

6.4 Results Summary

The research questions stated in section *1.2 - Main Research Questions* are here answered with summirized answers referring to the rest of the paper.

How shall camera images, maps and sensor data be presented in order to maximize the safety and efficiency of the operation? For this application it was found that the 360° video was not utilized to its full potential, see *6.3.1 - Driving Experience and Support Functions*. Also a rotating map was preferred to a fixed map with a rotating vehicle. The LiDAR drawn in the map described in section *5.3.3 - Maps* and section *5.3.5.1 - Light Detection And Ranging*, was found to work well.

At what level does the operator control the vehicle? As if sitting inside or are more high level commands (i.e. "Go to unloading location") issued? How do delays in the communication channel affect the choice of control? Because of the given implementation of the autonomous functions more high level commands could not be tested. However this is discussed in section *7 - Conclusion & Future Work*. It was found in this application that when driving manually 300 ms seconds was an acceptable latency. After this the operation became less fluent, see section *6.3.2 - Impact of Latency*.

Are there existing standards for remote control of autonomous vehicles? There exists standards relevant to teleoperation and autonomous control, mostly about testing methods which does not apply to this project. No standards for communication was found, but one of the proposed use cases for the fifth generation (5G) wireless systems is communication with and between autonomous vehicles, see section *4.8.3 - 5th Generation Wireless Systems* . More standards are discussed in section *6.1 - Standards and System Layout*.

How can the system be scalable to a variety of different sensors depending on application, and what are the requirements of the communication link for different types of sensors? By using a modular design where different functions can be added or removed depending on vehicle and application. In this application ROS has been used (see section *5.2 - Framework - Robot Operating*

System) and the modular design can be viewed in section *5.1 - System Architecture*. Depending on the amount of data that each sensor needs to transmit sets requirements for the communication channel. The cameras uses the majority of the bandwidth and depending on if the stitching process is computed onboard the vehicle or not different amount of data is needed. More can be read in section *4.8.4 - Link Requirement* and section *5.3.1 - Cameras*.

How will the vehicle switch between autonomous operation and manual control? What system has priority in different situations, what kind of handshakes are needed? For a safe and reliable transitions between autonomous operation and manual control the vehicle is needed to make a complete stop before changing state. More on how the synchronization is implemented in this application can be read in section *6.2 - Autonomous synchronization* . The operator always have the ability to take control of the vehicle at any state if necessary and override the stop handshake.

7

Conclusion & Future Work

In this thesis a prototype system for teleoperation has been developed, implemented and evaluated for a normally autonomous vehicle. Instead of the normal procedure of first remote controlling the vehicle, and gradually letting it perform autonomous functions, teleoperation has been added afterwards. This has given us the opportunity to design a system with manual takeover from autonomous control as primary use. Since the autonomous functions were present, the autonomous/manual synchronization was built around this system and its limitations. Since all autonomous functions are pre-recorded it is simple to return to the current autonomous task after a manual intervention because the path is always known. In a system where dynamic path planning is done there is room to create a more extensive manual intervention system. For instance marking preferred areas to drive or areas to avoid or drive around. This opens for lots of possibilities where the truck can be manually controlled in different ways, but not necessarily manually driven. It also makes the synchronization between manual and autonomous mode more complex because unlike this case it is not clear at all times what actually controls the vehicle.

Another simplifying factor in this application is that the paths do not overlap each other. Therefore it is always clear where in the path it is desired to resume. If the system is implemented on, for instance, an autonomous excavator, a recorded path of the bucket will most probably overlap itself many times. Using this resume approach would then yield a problem of where in the path the user wants to resume if placing the bucket in a place where multiple segments of the path meets.

The autonomous vehicle has extra sensors for navigation and obstacle detection such as LiDARs and GNSS. In addition, cameras are added for a surround view of the vehicle and stitched together to a full 360° view that the operator can pan in. On top of the video stream maps, offset to chosen autonomous path and the speed of the vehicle is overlaid. In this particular application, the usage of full camera surround view has not been utilized since the truck is mostly driven forward. One forward angled and one reverse angled camera would have been sufficient. However, this may not be the case when operating, for example, an excavator or a forest harvester which is often stationary and the work area is all around the vehicle. It would be interesting to use a head-mounted display with the camera surround view which we believe would utilize it better. It would allow for the driver to actually look around and mimic sitting inside the vehicle. In such case more cameras around the vehicle and not just the on cab would be beneficial to get a better 360° view.

One of the major difficulties in remote control and teleoperation is latency. Both in the initial literature study and in our evaluation it was found that 1000 ms seems to be the upper limit for operating a vehicle safely. However we believe that this is very application specific. Depending on the speed of movement and precision of the vehicle as well as the operational space, latencies are of differing importance. A large tanker ship on open sea or flying surveillance drones can handle higher delay times than an excavator or a mining truck in a narrow tunnel with preserved control. If latencies are too high for manual diving, it would be interesting to evaluate small commands of higher level manual control such as *"Reverse 20 meters"*.

It was also found that having small latencies rather then high frame rate was preferred. Lowering the frame rate and image quality would keep latencies low. An option to set these or by automatically analysing the connection and adjusting accordingly would probably benefit a system like this. Further investigation on haptic feedback in controls would be interesting if it is applicable in this type teleoperation. This requires though that latencies are kept small for it to function and actually aid the driver when in manual control.

The next major step with proposed system is to test it on a real construction truck to verify that the results from the simulation corresponds to reality.

Bibliography

[1] K. Hanson, "What's the difference between radio controlled and remote controlled vehicles?" *Toptenreviews*, 2010.

[2] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006.

[3] S. Thrun, "Toward robotic cars," *Commun. ACM*, vol. 53, no. 4, pp. 99–106, Apr. 2010.

[4] R. Carelli, G. Forte, L. Canali, V. Mut, G. Araguás, and E. Destéfanis, "Autonomous and teleoperation control of a mobile robot," *Mechatronics*, vol. 18, no. 4, pp. 187 – 194, 2008.

[5] VolvoGroup, "The future of automation is happening now at Volvo," 2016, accessed: 2016-05-12. [Online]. Available: http://news.volvogroup.com/2016/05/09/the-future-of-automation-is-happening-now-at-volvo/

[6] S. J. Velat, J. Lee, N. Johnson, and C. D. Crane, "Vision based vehicle localization for autonomous navigation," in *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, June 2007, pp. 528–533.

[7] Caterpillar Inc., "Cat command for dozing," 2012, accessed: 2016-05-10. [Online]. Available: http://s7d2.scene7.com/is/content/Caterpillar/C10338821

[8] Caterpillar Inc., "Cat command for hauling," 2013, accessed: 2016-05-10. [Online]. Available: http://s7d2.scene7.com/is/content/Caterpillar/C10338825

[9] Caterpillar Inc., "Cat command for underground," 2013, accessed: 2016-05-10. [Online]. Available: http://s7d2.scene7.com/is/content/Caterpillar/C10338828

[10] Komatsu, "Service and solutions," accessed: 2016-05-10. [Online]. Available: http://www.komatsu.com/CompanyInfo/profile/product_supports/

[11] Hitachi, "Hitachi to develop autonomous haulage system," accessed: 2016-05-10. [Online]. Available: http://www.hcme.com/Our-Company/News/Press-releases/Hitachi-to-develop-Autonomous-Haulage-System

[12] Sandvik, "Automine hauling," accessed: 2016-05-10. [Online]. Available: http://mining.sandvik.com/en/products/equipment/mine-automation-systems/equipment-automation/automine-hauling

[13] Sandvik, "Automine loading," accessed: 2016-05-10. [Online]. Available: http://mining.sandvik.com/en/products/equipment/mine-automation-systems/equipment-automation/automine-loading

[14] Sandvik, "Automine surface drilling," accessed: 2016-05-10. [Online]. Available: http://mining.sandvik.com/en/products/equipment/mine-automation-systems/equipment-automation/automine-surface-drilling

[15] AtlasCopco, “Scooptram automation: Underjordisk last-maskin,” accessed: 2016-05-10. [Online]. Available: http://mining.sandvik.com/en/products/equipment/mine-automation-systems/equipment-automation/automine-surface-drilling

[16] AtlasCopco, “Multipel effektivitet från ett säkert avstånd med Atlas Copcos nya BenchREMOTE,” accessed: 2016-05-10. [Online]. Available: http://www.atlascopco.se/sesv/news/productnews/141218_multiplied_efficiency_from_a_safe_distance_with_atlas_copcos_new_benchremote.aspx

[17] Caterpillar Inc., “Cat Minestar,” accessed: 2016-05-10. [Online]. Available: http://www.cat.com/en_US/support/operations/technology/cat-minestar/command.html

[18] Remoquip, “Universal remote control for hydraulic equipment,” accessed: 2016-05-10. [Online]. Available: http://remoquip.com/

[19] Autonomous Solutions, Inc, “ASI Robotics,” accessed: 2016-05-10. [Online]. Available: https://www.asirobots.com/

[20] Hard-Line, “Hard-Line Teleoperation,” accessed: 2016-05-10. [Online]. Available: http://www.hard-line.com/en/

[21] Torc Robotics, “Autonomous Systems,” accessed: 2016-05-10. [Online]. Available: http://www.torcrobotics.com/autonomy/

[22] AEE, “Autonomous Earthmoving Equipment LLC,” accessed: 2016-05-10. [Online]. Available: http://aee.us.com/

[23] Taurob, “Universal teleoperation control,” accessed: 2016-05-10. [Online]. Available: http://taurob.com/

[24] 3D Simulators for Construction Equipment, “Universal teleoperation control,” accessed: 2016-05-10. [Online]. Available: http://www.oryx.se/

[25] D. D. Woods, “Visual momentum: a concept to improve the cognitive coupling of person and computer,” *International Journal of Man-Machine Studies*, vol. 21, no. 3, pp. 229 – 244, 1984.

[26] D. D. Woods, J. Tittle, M. Feil, and A. Roesler, “Envisioning human-robot coordination in future operations,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 2, pp. 210–218, 2004.

[27] P. L. Alfano and G. F. Michel, “Restricting the field of view: perceptual and performance effects,” *Perceptual and motor skills*, vol. 70, no. 1, p. 35, 1990.

[28] J. B. Van Erp and P. Padmos, “Image parameters for driving with indirect viewing systems,” *Ergonomics*, vol. 46, no. 15, pp. 1471–1499, 2003.

[29] J. Y. C. Chen, E. C. Haas, and M. J. Barnes, “Human performance issues and user interface design for teleoperated robots,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1231–1245, 2007.

[30] J. Corde Lane, C. Carignan, B. Sullivan, D. Akin, T. Hunt, and R. Cohen, “Effects of time delay on telerobotic control of neutral buoyancy vehicles,” in *Robotics and Automation, 2002. Proceedings. ICRA ’02. IEEE International Conference on*, vol. 3, 2002, pp. 2874–2879.

[31] R. R. Mourant and T. R. Thattacherry, "Simulator sickness in a virtual environments driving simulator," *Proceedings of the Human Factors and Ergonomics Society ...Annual Meeting*, vol. 1, p. 534, 2000.

[32] J. Wallner, T. Tang, and M. Lienkamp, "Development of an emergency braking system for teleoperated vehicles based on lidar sensor data," in *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, vol. 02, Sept 2014, pp. 569–576.

[33] J. Lister, "What is the theoretical maximum throughput for wi-fi?" accessed: 2016-03-15. [Online]. Available: http://classroom.synonym.com/theoretical-maximum-throughput-wifi-18050.html

[34] Netgear, "What is the difference between 2.4 ghz and 5ghz?" 2015, accessed: 2016-03-15. [Online]. Available: http://kb.netgear.com/app/answers/detail/a_id/29396/~/what-is-the-difference-between-2.4-ghz-and-5ghz%3F

[35] J. Bertolucci, "Six things that block your wi-fi, and how to fix them," accessed: 2016-03-16. [Online]. Available: http://www.pcworld.com/article/227973/six_things_that_block_your_wifi_and_how_to_fix_them.html

[36] B. Mitchell, "What is the range of a typical wi-fi network?" accessed: 2016-03-17. [Online]. Available: http://compnetworking.about.com/cs/wirelessproducts/f/wifirange.htm

[37] P. Belanger, "802.11n delivers better range," 2007, accessed: 2016-03-17. [Online]. Available: http://www.wi-fiplanet.com/tutorials/article.php/3680781

[38] CWNP, "Wi-fi overhead, part 1: Sources of overhead," 2011, accessed: 2016-03-17. [Online]. Available: https://www.cwnp.com/wi-fi-overhead-part-1-sources-of-overhead/

[39] J. E. Rachid El Hattachi, "Ngmn 5g white paper," 2015, accessed: 2016-05-10. [Online]. Available: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf

[40] E. Press., "5g pilot launched to make swedish mines safer and more productive," 2015, accessed: 2016-05-10. [Online]. Available: http://www.ericsson.com/news/1924016

[41] P. de Bruin, "5g mining project goes underground," 2015, accessed: 2016-05-10. [Online]. Available: http://www.ericsson.com/research-blog/5g/5g-mining-project-goes-underground/

[42] C. Nielsen, M. Goodrich, and R. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 927–941, Oct 2007.

[43] L. C. Thomas and C. D. Wickens, "Visual displays and cognitive tunneling: Frames of reference effects on spatial judgments and change detection," *Proceedings of the Human Factors and Ergonomics Society ...Annual Meeting*, vol. 1, pp. 336–340, 2001.

[44] A. Zelinsky, "Learning opencv—computer vision with the opencv library (bradski, g.r. et al.; 2008)[on the shelf]," *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 100–100, 2009.

[45] K. Kruckel, F. Nolden, A. Ferrein, and I. Scholl, "Intuitive visual teleoperation for ugvs using free-look augmented reality displays," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4412–4417.

[46] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 08 2007.

[47] E. Laurier and B. Brown, "Rotating maps and readers: praxiological aspects of alignment and orientation," *Transactions of the Institute of British Geographers*, vol. 33, no. 2, pp. 201–216, 2008.

[48] P. Shah and A. Miyake, *The Cambridge Handbook of Visuospatial Thinking*, ser. Cambridge Handbooks in Psychology. Cambridge University Press, 2005.

[49] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, June 2006.

[50] A. Liu and S. Chang, "Force feedback in a stationary driving simulator," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, vol. 2, Oct 1995, pp. 1711–1716 vol.2.

[51] J. L. Marble, D. J. Bruemmer, and D. A. Few, "Lessons learned from usability tests with a collaborative cognitive workspace for human-robot teams," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 448–453 vol.1.

[52] N. O. Aykut, E. Gülal, and B. Akpinar, "Performance of single base rtk gnss method versus network rtk," *Earth Sciences Research Journal*, vol. 19, no. 2, p. 135, 2015.

[53] W. A. Aviles, T. W. Hughes, H. R. Everett, A. Y. Umeda, S. W. Martin, A. H. Koyamatsu, M. R. Solorzano, R. T. Laird, and S. P. McArthur, "Issues in mobile robotics: the unmanned ground vehicle program teleoperated vehicle," in *Mobile Robots V*, vol. 1388, 1991.

[54] M. Lewis, J. Wang, S. Hughes, and X. Liu, "Experiments with attitude: attitude displays for teleoperation," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 2, Oct 2003, pp. 1345–1349 vol.2.

[55] Volvo Trucks, "Data sheet Volvo FMX13 8x4," accessed: 2016-05-23. [Online]. Available: http://segotn12827.rds.volvo.com/STPIFiles/Volvo/ModelRange/fm84fr3cbx_gbr_eng.pdf

[56] ANSI, "Standard test method for evaluating emergency response robot capabilities: Human-system interaction (hsi): Search tasks: Random mazes with complex terrain," American National Standards Institute, Washington, DC, ASTM E2853-12, 2012.

[57] ISO, "Earth-moving machinery – safety requirements for remote operator control systems," International Organization for Standardization, Geneva, Switzerland, ISO 15817, 2012.