# Requirement Specification

Autonomous Truck with Trailer

December 15, 2022

Version 1.1

ADAPT™

Status

| | | |
|---|---|---|
| Reviewed | Alfred Sundstedt | 2022-09-20 |
| Approved | Shamisa Shoja | 2022-09-21 |



ADAPT
alfsu259@liu.se

# Project Identity

Group E-mail: alfsu259@liu.se

Homepage: https://www.control.isy.liu.se/student/tsrt10/

Orderer: Shamisa Shoja, Reglerteknik, ISY
E-mail: shamisa.shoja@liu.se

Customer: Daniel Axehill, Reglerteknik, ISY
E-mail: daniel.axehill@liu.se

Supervisor: Carl Hynén Ulfsjöö, Reglerteknik, ISY
E-mail: carl.hynen@liu.se

Course Responsible: Daniel Axehill, Reglerteknik, ISY
E-mail: daniel.axehill@liu.se

## Participants of the group

| **Name** | **Responsibility** | **E-mail** |
|---|---|---|
| Martin Axelsson | | marax633@student.liu.se |
| Jesper Barreng | Test Manager | jesba281@student.liu.se |
| Isak Bokne | Design Manager | isabo438@student.liu.se |
| Charlie Elf | | chael086@student.liu.se |
| Terese Johansson | Document Manager | terjo233@student.liu.se |
| Alfred Sundstedt | Project Leader | alfsu259@student.liu.se |
| Emil Wiman | Software Architect | emiwi425@student.liu.se |

# CONTENTS

# DOCUMENT HISTORY

| Version | Date | Changes made | Sign | Reviewer |
|---|---|---|---|---|
| 0.1 | 2022-09-16 | First draft | C.Elf, E.Wiman, I.Bokne | A.Sundstedt |
| 0.2 | 2022-09-20 | Second draft | C.Elf, E.Wiman, I.Bokne | M.Axelsson |
| 1.0 | 2022-09-21 | Final draft. Changed priority of requirement 3 | C.Elf, E.Wiman, I.Bokne | A.Sundstedt |
| 1.1 | 2022-11-16 | Revised according to renegotiation with orderer. Changed requirements 3, 8, 10, 18, 19, 20 and 47. | E.Wiman | A.Sundstedt |

# 1 INTRODUCTION

This document states the requirements for the project "Autonomous truck with trailer" in the automatic control CDIO project course TSRT10 at Linköping university. This project is conducted during the fall semester of 2022.

The foundation of the project has been built by prior years' project groups. The documentation from the previous group can be found here [1]. The focus of this year's project is to develop a planning algorithm which can take uncertainty into account when planning trajectories in an environment with dynamic obstacles. Moreover, models for predicting pedestrians and other ground vehicles should be developed. This will in turn require a considerable change in the software architecture and in the visualization system of the truck. The existing LEGO truck can be seen in Figure 1.

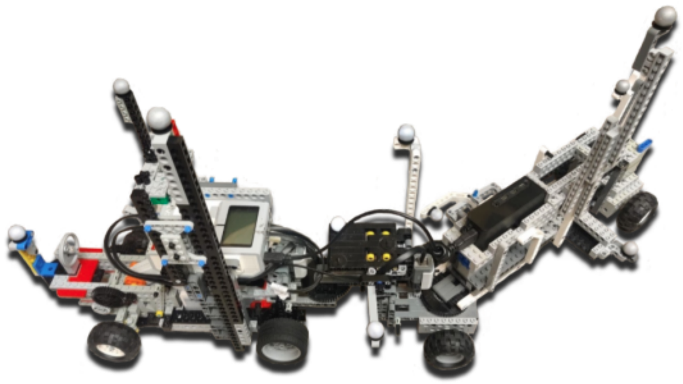

**Figure 1:** The existing truck system with the truck, dolly and trailer.

## 1.1 Partners

- The project group consists of seven Master degree students studying at Linköping University
- Carl Hynén Ulfsjöö, project supervisor
- Shamisa Shoja, project orderer
- Daniel Axehill, examiner and customer

## 1.2 Background information

The field of research regarding autonomous vehicles has been on the uprise during the last decade. For example, cars are becoming more and more autonomous, with features such as automatic parking and autopilot. Taking dynamic obstacles into consideration when planning and driving is essential to avoid hazardous situations. Research in this area can hopefully lead to smarter and more robust systems which can lay a foundation for future autonomous vehicles.

### 1.3 Aims and goals

The purpose of this project is to investigate and implement algorithms for autonomous maneuvering of a truck in a complex dynamic environment. This year's work will be built upon earlier groups work as mentioned earlier. The main focus of this year can be summarized as:

- Investigation and implementation of planning algorithms that can handle dynamic obstacles with uncertainty
- Investigation of models used for prediction of pedestrians and other dynamic obstacles
- Development of the systems architecture to implement the new functionality and to enable easier development for future groups
- Development of the truck's visualization system to illustrate its surroundings and current path

The long-term aim of the project is to develop a robust system that can be used for research and education at the Department of Electrical Engineering.

### 1.4 Definition of terms

The abbreviations used in this document can be found in the bullet list below.

- **Git** - A version control tool used in software development
- **RPi** – **R**aspberry **Pi**, a single board computer
- **ROS** - **R**obot **O**perating **S**ystem, an open source framework for robot software
- **Visionen** - An arena for research and education at Linköping University
- **QualiSys** - A positioning system used in Visionen
- **POMDP** - **P**artially **O**bservable **M**arkov **D**ecision **P**rocesses, A mathematical framework for decision making with uncertainty. The agent can not observe the full underlying state, hence it is partially observable
- **RViz** - a visualization tool used in ROS.
- **IMM filter** - **I**nteractive **M**otion **M**odel filter is a filter designed to track several objects that are highly maneuverable.
- **MPC** - **M**odel **P**redictive **C**ontroll is an optimization based control method.

### 1.5 Requirement definition

This section presents the format of a requirement, see the example below. In the leftmost column is the requirement number given, here represented by *X*. The version column contains information regarding if the requirement was an original one or if it has been revised. The description column describes the requirement. The priority column describes the priority of the requirement, where priority 1 is a mandatory requirement while priority 2 or 3 should be done if excess time is available.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| X | Original | Requirement description | 1,2,3 |

# 2 SYSTEM OVERVIEW

## 2.1 Description of the system

This projects main system is based around a LEGO truck with a trailer. It includes a LEGO EV3 and an RPi unit for computing. A project laptop with the framework for robot software ROS will be used for simulating and operating the truck. The truck is equipped with small reflectors for positioning in the visualization area Visionen. Last years system was controlled using an MPC controller. An illustration of the complete existing system can be seen in Figure 2. [1] In this figure the different sub-systems are presented as blocks and the lines between them describe the data flows. The system is given a start and goal node from the user. The planner plans a path in the static environment with data from the state observer. This plan is then handed to the controller. The controller produces control signals which are sent to the simulations or the actual truck for execution. The state observer also takes in the control signals and uses them to update the state. The predictor in the current systems affects the control signals to avoid dynamic obstacles. The state observer also communicates with the simulator or Qualisys to gather data connected to the world.

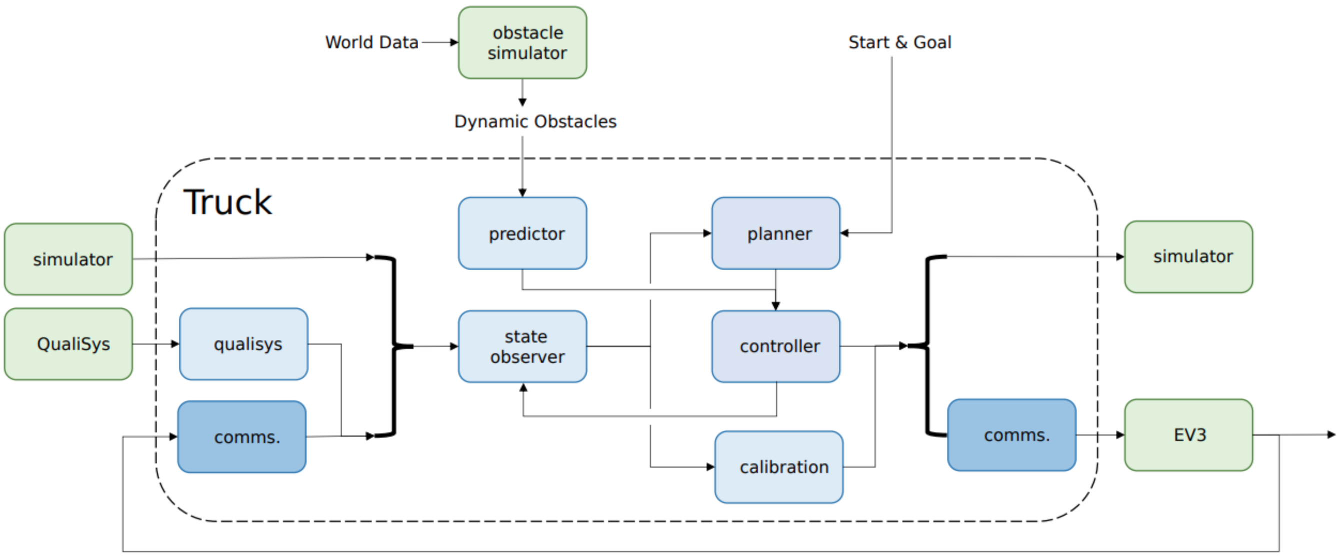


**Figure 2:** An overview of the existing system.

The main objective for this project is to alter this system architecture to handle dynamic obstacles. The new control system is similar to the previous one, see Figure 3. The main difference is that the whole system must be time-dependent in order to handle dynamic obstacles. This can be done by introducing time as a state to the system.

Moreover, the motion planner should receive feedback from a predictor in order to predict dangerous future states which it needs to avoid. The aim is to alter the motion planner so it utilizes *Partially Observable Markov Decision Process* to plan with the given uncertainty in future states. To predict future objects and their motion is the aim to implement an *IMM filter* in the predictor block.

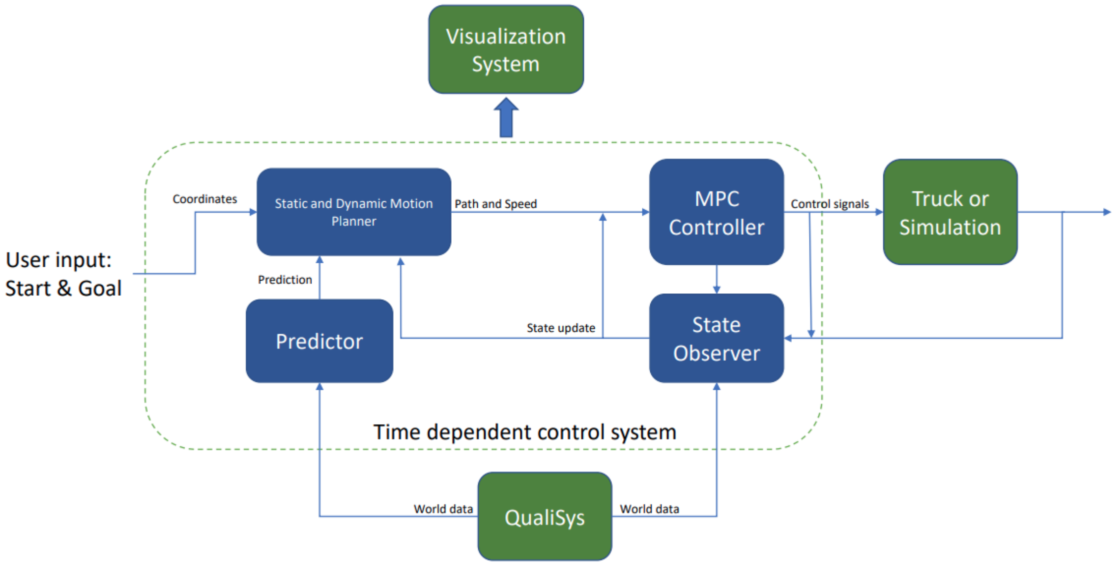


**Figure 3:** An overview of the future control system.

### 2.2 Included sub-systems

The system is essentially consisting of four sub-systems including:

- A LEGO truck with a trailer controlled using the output signal from the MPC controller
- Control system with motion planning (both static and dynamic), a controller (MPC-controller) and a state observer
- A simulation environment using RViz with ROS to test implementations without the actual LEGO truck
- Visionen visualization area with positioning system

### 2.3 Limitations

Project limitations include:

- The operating area is flat
- The existing truck hardware will not be expanded

- Preinstalled software such as ROS 1
- Limited time in Visionen test area
- Dependent of Visionen being functional as it is the only test site available
- The project must be finished in December 2022

### 2.4 Design philosophy

As the project is ongoing for students at Linköping University, the aim is to keep the work organized and structured so that new people to the project can understand and get going fairly quick. During the project it is also of importance to have frequent tests in order to make sure that new implementations are working properly.

### 2.5 General requirements

Below, the general requirements for the goal of the project will be presented.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 1 | Original | Requirements that the project group are unable to fulfill should be renegotiated with the orderer well before the deadline | 1 |
| 2 | Original | The obstacles should be able to be placed arbitrary | 1 |
| 3 | Revised 2022-11-16 | No computations should be done by an external computer | 3 |
| 4 | Original | The system should have an architecture so that it will be easy for future groups to develop new functionality | 1 |

## 3 LEGO TRUCK

The physical truck has some weaknesses. The probes for mounting the reflectors are not very stable and may sway while the system is running, causing the Visionen sub-system to lose track of the body. The truck also needs to be weighed down to make sure that all wheels are in contact with the ground at all times. This indicates that there could be tension in the bodies which impedes the wheels from full ground contact.

### 3.1 Physical requirements

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 5 | Original | The reflectors should be rigidly mounted such that their maximum deviation is neglible in regards to localization | 1 |
| 6 | Original | There should always be good contact between the wheels and the ground | 3 |

# 4 CONTROL SYSTEM

The LEGO truck controller contains mainly three parts. A state observer, a motion planner and an MPC controller. These three will work together to make the LEGO truck go from a starting point to a goal point along a planned path while avoiding static and dynamic obstacles.

- State observer - QualiSys, the built-in positioning system in Visionen, gives the position coordinates $(x, y)$ and the rotation $(\theta)$ of each of the three bodies constituting the LEGO truck. These positions and rotations constitute the current state as $(x_{truck}, y_{truck}, \theta_{truck}, x_{dolly}, y_{dolly}, \theta_{dolly}, x_{trailer}, y_{trailer}, \theta_{trailer})$. For dynamic obstacles the system will also provide the states as they are observed by the QualiSys system.
- Motion planner - Contains a static and a dynamic motion planner. The static motion planner uses predefined obstacles to generate the path. The dynamic motion planner uses the state of the LEGO truck and the states of dynamic obstacles to continuously update the trajectory. This will be done using POMDP.
- MPC controller - Uses the motion plan to generate signals for the LEGO truck such that it can progress towards the goal state without deviating from the planned trajectory and driving into obstacles.
- Predictor - The purpose of the motion planner is to predict future states for dynamic objects. The aim is to implement this predictor using an IMM filter.

In the following sections different requirements for the controller will be presented.

## 4.1 Design requirements

Below follows the design requirements for the control system. These requirements corresponds to the illustration of the time dependent control system in Figure 3.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 7 | Original | The control system should be time-dependent | 1 |
| 8 | Revised 2022-11-16 | The control system must be implemented on the RPi | 3 |
| 9 | Original | The controller must work in real-time on the RPi | 1 |
| 10 | Revised 2022-11-16 | The motion planner should work in real time on the RPi | 3 |

*cont. on next page*

| *cont. from previous page* | | | |
| --- | --- | --- | --- |
| Requirement | Version | Description | Priority |
| 11 | Original | The implemented code should utilize the ROS framework | 1 |
| 12 | Original | The end user should be able to change the control parameters online | 1 |
| 13 | Original | The motion planner should implement POMDP to handle uncertainty when planning | 1 |

### 4.2 Functional requirements for the control system

Below follows the functional requirements for the control system.

| Requirement | Version | Description | Priority |
| --- | --- | --- | --- |
| 14 | Original | The control system will plan a feasible trajectory from a start state to a goal state | 1 |
| 15 | Original | The dynamic motion planner uses the path from the static motion planner to create a trajectory | 1 |
| 16 | Original | The dynamic motion planner should be able to calculate a new path that deviates from the original path if the risk of collision is above some threshold | 2 |
| 17 | Original | The planned trajectory will always avoid any static obstacles | 1 |
| 18 | Revised 2022-11-16 | The predictor can predict the trajectory of 1 dynamic obstacle which is within a 3 m radius from the rear axle | 1 |
| 19 | Revised 2022-11-16 | The predictor can predict the trajectory of 3 dynamic obstacles which are within a 3 m radius from the rear axle | 2 |
| 20 | Revised 2022-11-16 | The predictor can predict the trajectory of 5 dynamic obstacles which are within a 3 m radius from the rear axle | 3 |
| 21 | Original | The motion planner will continuously update the planned trajectory to avoid dynamic obstacles | 1 |
| 22 | Original | The control system will take longitudinal action to avoid dynamic obstacles | 1 |
| 23 | Original | The control system will take lateral action to avoid dynamic obstacles | 2 |
| 24 | Original | The truck should be able to finish an entire mission, that includes driving forward, reversing and avoidance of static obstacles, without deviating from the generated path with more than 15 cm | 1 |

### 4.3 Interfaces

The control system works simultaneously with other sub-systems. The requirements of the interfaces are described here.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 25 | Original | All components of the control system should use ROS to communicate with other subsystems | 1 |
| 26 | Original | The motion planner should be able to send the planned path to other subsystems, mainly for visualization | 1 |
| 27 | Original | The predictor should receive information from the QualiSys system | 1 |
| 28 | Original | The motion planner should consider information from the predictor when planning | 1 |

# 5 PERFORMANCE REQUIREMENTS

In this section the performance requirements for the entire system is specified. Regarding requirements 29 and 30, these correspond to requirements 22 and 23. If the truck should be able to reach its goal state when a dynamic obstacle comes to a permanent stop in its path (i.e. fulfilling requirement 30), it needs to be able to take lateral action to avoid the dynamic obstacle (and thus satisfying requirement 23).

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 29 | Original | The truck reaches its goal state within a feasible time unless a dynamic obstacle comes to a permanent stop in the path of the truck | 1 |
| 30 | Original | The truck reaches its goal state within a feasible time when a dynamic obstacle comes to a permanent stop in the path of the truck | 2 |
| 31 | Original | The truck will keep a distance of at least 10 cm to static obstacles | 1 |
| 32 | Original | The truck will keep a distance of at least 10 cm to dynamic obstacles in scenarios where the dynamic obstacle does not approach the truck after the truck has become stationary | 1 |

# 6 VISUALIZATION

The purpose of the visualization system is to project the environment in which the truck will execute its missions. The visualization system should also be able to project the truck's path. This will in turn give us a better understanding of how the system performs in real life and not only in simulation. Moreover, it is a good way to show the potential of the truck to external actors.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 33 | Original | The visualization software connects with the RPi | 1 |
| 34 | Original | The visualization system should run on the computer in Visionen connected to the projector | 1 |
| 35 | Original | The visualization should display the initial planned path | 1 |
| 36 | Original | The visualization should display the path driven during and after the mission | 1 |
| 37 | Original | The visualization should display the static objects | 1 |
| 38 | Original | The visualization should display the dynamic objects | 1 |
| 39 | Original | The visualization should display changes in the path that occurred due to dynamic obstacles | 1 |

# 7 ECONOMY

Economy for this project comes in the form of some resources with requirements as below.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 40 | Original | Each group member will spend 240 hours on the project | 1 |
| 41 | Original | The group will receive 40 hours of tutoring time | 1 |
| 42 | Original | The group has a project room to use | 1 |
| 43 | Original | Hardware for the project is provided by the customer | 1 |

# 8 DELIVERY

Below follows the requirements for all tollgates (TG).

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 44 | Original | TG2: Requirement specification, project plan (including time plan) and a design specification draft. Date: 2022-09-21 | 1 |
| 45 | Original | TG3: Design specification and test plan. Date: 2022-10-12 | 1 |
| 46 | Original | TG4: Draft of deliveries for TG5. Date: 2022-11-24 | 1 |
| 47 | Revised 2022-11-16 | TG5: All functionality, test protocol, user manual, demonstration of fulfilled requirements. Date: 2022-12-02 | 1 |
| 48 | Original | TG6: Technical report, poster, website, demo film. Date: 2022-12-19 | 1 |

# 9 DOCUMENTATION

Table 11 lists all documents that shall be produced in the project. Here PG denotes *Project Group*, O denotes *Orderer* and C denotes *Customer*.

**Table 11:** Documents to be produced.

| Document | Aim | Target | Format |
|---|---|---|---|
| Project plan | Describes how the project will be conducted as well as a time plan with activities and tollgates | PG, O | PDF |
| Requirement specification | Describes the requirements on the workflow and result of the project. | PG, O, C | PDF |
| Design specification | Describes and illustrates the whole system in detail. | PG, O | PDF |
| Test plan | Describes what tests are conducted to measure how well requirements are fulfilled. | O | PDF |
| Test protocol | Contains the results of the test conducted. | O | |
| User manual | A guide on how to use the system. | C | PDF |
| Technical report | A report on the technical results of the project. | O | PDF |
| Time report | Time sheet on how time was distributed. | O | Excel sheet |
| Meeting protocol | Contains the agenda and decisions made during meetings. | PG | PDF |
| After study | Describes how the project proceeded, e.g. teamwork, workload etc. | C | PDF |

# 10 SOFTWARE QUALITY

Below follows the requirements for all software produced in this project.

| Requirement | Version | Description | Priority |
|---|---|---|---|
| 49 | Original | Code should be well written and follow Googles style guide [2] | 1 |
| 50 | Original | Code should be documented | 1 |
| 51 | Original | Gitlab should be used for version control | 1 |
| 52 | Original | A separate branch should be made for this project | 1 |
| 53 | Original | A merge request can either be approved by only the software responsible or by two other project members | 1 |
| 54 | Original | Implemented subsystems should have a corresponding test it must complete before it can be pushed to the main branch | 2 |

# REFERENCES

[1] J. Rosengren, E. Sellén, D. Larsson, D. Similä, G. Ingemarsson, J. Sjöblom, and O. Bergström. (2021) Technical documentation, autonomous truck with a trailer. [Online]. Available: http://www.isy.liu.se/edu/projekt/tsrt10/2021/rev_truck/

[2] Google. (2022) Google style guide. [Online]. Available: https://google.github.io/styleguide/