Examensarbete

Motion Planning and Stabilization for a Reversing Truck and Trailer System

Examensarbete utfört i reglerteknik vid Tekniska högskolan vid Linköpings universitet av

Oskar Ljungqvist

LiTH-ISY-EX--15/4879--SE

Linköping 2015
Motion Planning and Stabilization for a Reversing Truck and Trailer System

Examensarbete utfört i reglerteknik vid Tekniska högskolan vid Linköpings universitet av

Oskar Ljungqvist

LiTH-ISY-EX--15/4879--SE

Handledare: Niclas Evestedt
ISY, Linköpings universitet

Examinator: Daniel Axehill
ISY, Linköpings universitet

Linköping, 17 juni 2015
Banplanering och stabilisering av en backande lastbil med släpvagn

Motion Planning and Stabilization for a Reversing Truck and Trailer System

Oskar Ljungqvist

This thesis work contains a stabilization and a motion planning strategy for a truck and trailer system. A dynamical model for a general 2-trailer with two rigid free joints and a kingpin hitching has been derived based on previous work. The model holds under the assumption of rolling without slipping of the wheels and has been used for control design and as a steering function in a probabilistic motion planning algorithm.

A gain scheduled Linear Quadratic (LQ) controller with a Pure pursuit path following algorithm has been designed to stabilize the system around a given reference path. The LQ controller is only used in backward motion and the Pure pursuit controller is split into two parts which are chosen depending on the direction of motion.

A motion planning algorithm called Closed-Loop Rapidly-exploring Random Tree (CL-RRT) has then been used to plan suitable reference paths for the system from an initial state configuration to a desired goal configuration with obstacle-imposed constraints. The motion planning algorithm solves a non-convex optimal control problem by randomly exploring the input space to the closed-loop system by performing forward simulations of the closed-loop system.

Evaluations of performance is partly done in simulations and partly on a Lego platform consisting of a small-scale system. The controllers have been used on the Lego platform with successful results. When the reference path is chosen as a smooth function the closed-loop system is able to follow the desired path in forward and backward motion with a small control error.

In the work, it is shown how the CL-RRT algorithm is able to plan non-trivial maneuvers in simulations by combining forward and backward motion. Beyond simulations, the algorithm has also been used for planning in open-loop for the Lego platform.
Abstract

This thesis work contains a stabilization and a motion planning strategy for a truck and trailer system. A dynamical model for a general 2-trailer with two rigid free joints and a kingpin hitching has been derived based on previous work. The model holds under the assumption of rolling without slipping of the wheels and has been used for control design and as a steering function in a probabilistic motion planning algorithm.

A gain scheduled Linear Quadratic (LQ) controller with a Pure pursuit path following algorithm has been designed to stabilize the system around a given reference path. The LQ controller is only used in backward motion and the Pure pursuit controller is split into two parts which are chosen depending on the direction of motion.

A motion planning algorithm called Closed-Loop Rapidly-exploring Random Tree (CL-RRT) has then been used to plan suitable reference paths for the system from an initial state configuration to a desired goal configuration with obstacle-imposed constraints. The motion planning algorithm solves a non-convex optimal control problem by randomly exploring the input space to the closed-loop system by performing forward simulations of the closed-loop system.

Evaluations of performance is partly done in simulations and partly on a Lego platform consisting of a small-scale system. The controllers have been used on the Lego platform with successful results. When the reference path is chosen as a smooth function the closed-loop system is able to follow the desired path in forward and backward motion with a small control error.

In the work, it is shown how the CL-RRT algorithm is able to plan non-trivial maneuvers in simulations by combining forward and backward motion. Beyond simulations, the algorithm has also been used for planning in open-loop for the Lego platform.
Acknowledgments

First of all I will start by expressing my gratitude to my examiner Daniel Axehill and supervisor Niclas Evestedt for providing me with the opportunity to perform this master’s thesis work. I would also like to thank them for all the help and support I have got from them during the project. They have always been available for discussions regarding control strategies or solutions regarding implementation problems.

Without Niclas Evestedt’s implemented CL-RRT algorithm I would never have gone this far with this thesis work. The foundation to all contributed results regarding motion planning is because of his research work. For this I am truly grateful.

Finally, I also want to take the opportunity to show my gratitude to my family and friends for their shown interest and continuous support.

Linköping, June 2015
Oskar Ljungqvist
## Contents

Notation ix

1 Introduction 1
  1.1 Background and motivation 1
  1.2 Problem formulation 1
  1.3 Related work 2
  1.4 Outline 3

2 Preliminaries 5
  2.1 Linearization of a nonlinear model 5
  2.2 Linear Quadratic control 6
  2.3 Constrained motion 7

3 Lego platform 9
  3.1 Servo motors 9
  3.2 Steering mechanism 10
  3.3 Measurement system 10
    3.3.1 Angular sensors 10
    3.3.2 Position system using IR-cameras 11

4 Modeling 13
  4.1 Recursive formula based on kinematic constraints 14
  4.2 Derivation of nonlinear model for a general 2-trailer system 15
  4.3 Linearization along arc circles 17
  4.4 Modeling steering angle dynamics in simulations 18

5 Design of closed-loop system 21
  5.1 Derivation of the pure pursuit control law 22
  5.2 Closed-loop system in forward motion 23
  5.3 Closed-loop system in backward motion 24
    5.3.1 Pure pursuit controller in backward motion 24
    5.3.2 Feedback controller 25
    5.3.3 Gain scheduled feedback gain 25
5.3.4 Nonlinear pre-compensation link in backward motion . . . 27
5.4 Design and simulation results . . . . . . . . . . . . . . . . . . . . 28
5.5 Stability analysis of the closed-loop system . . . . . . . . . . . . 30

6 Motion planner 33
6.1 The motion planning problem . . . . . . . . . . . . . . . . . . . . 33
6.2 Planner . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
  6.2.1 Constraints on the final configuration . . . . . . . . . . . . 34
  6.2.2 Simulation of closed-loop system . . . . . . . . . . . . . . . 34
  6.2.3 Speed control . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
  6.2.4 Sampling Strategies . . . . . . . . . . . . . . . . . . . . . . . 35
  6.2.5 Objective function . . . . . . . . . . . . . . . . . . . . . . . 36
  6.2.6 Obstacle Map . . . . . . . . . . . . . . . . . . . . . . . . . . 36
  6.2.7 Heuristics . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
  6.2.8 The CL-RRT algorithm . . . . . . . . . . . . . . . . . . . . . 37

7 Results 41
7.1 Reference tracking of the closed-loop system . . . . . . . . . . . 41
7.2 Evaluation of CL-RRT algorithm in simulations . . . . . . . . . 44
7.3 Geometric sensitivity analysis . . . . . . . . . . . . . . . . . . . . 46
7.4 Geometric robustness analysis . . . . . . . . . . . . . . . . . . . . 48
7.5 Autonomous parking of the Lego platform . . . . . . . . . . . . . 49

8 Conclusions 53
8.1 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
8.2 Future work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

Bibliography 57
## Notation

### Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>Length of the truck</td>
</tr>
<tr>
<td>$M_1$</td>
<td>Length of the off-hitch</td>
</tr>
<tr>
<td>$(x_1, y_1)$</td>
<td>Mid position of the rear axle of the truck in global coordinates</td>
</tr>
<tr>
<td>$L_2$</td>
<td>Length of the dolly</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>Relative orientation of the truck with respect to the dolly</td>
</tr>
<tr>
<td>$(x_2, y_2)$</td>
<td>Mid position of the rear axle of the dolly in global coordinates</td>
</tr>
<tr>
<td>$L_3$</td>
<td>Length of the trailer</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>Relative orientation of the dolly with respect to the trailer</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>Absolute orientation of the trailer with respect to global coordinates</td>
</tr>
<tr>
<td>$(x_3, y_3)$</td>
<td>Mid position of the rear axle of the trailer in global coordinates</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Steering angle of the truck</td>
</tr>
<tr>
<td>$v$</td>
<td>Velocity of the truck’s rear axle</td>
</tr>
<tr>
<td>$R$</td>
<td>Look ahead distance, a design parameter for the Pure pursuit controller</td>
</tr>
<tr>
<td>$r_0$</td>
<td>Offset in sample radius</td>
</tr>
<tr>
<td>$\sigma_{r_0}$</td>
<td>Standard deviation in sample radius</td>
</tr>
<tr>
<td>$\theta_0$</td>
<td>Offset in sample angle</td>
</tr>
<tr>
<td>$\sigma_{\theta_0}$</td>
<td>Standard deviation in sample angle</td>
</tr>
</tbody>
</table>
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>LQ</td>
<td>Linear Quadratic controller</td>
</tr>
<tr>
<td>RRT</td>
<td>Randomly-exploring Random Tree</td>
</tr>
<tr>
<td>RRT*</td>
<td>Randomly-exploring Random Tree Star</td>
</tr>
<tr>
<td>CL-RRT</td>
<td>Closed-Loop Randomly-exploring Random Tree</td>
</tr>
</tbody>
</table>
The work in this master thesis concerns modeling, controlling, implementing and motion planning of a truck and trailer system. According to the theory of vehicle control, the system is a general 3-trailer [6] with two rigid free joints and a kingpin hitching. In this chapter we will discuss the underlying background, related work and motivation why this area of automatic control is interesting. The problem formulation and the outline of the thesis will also be presented.

1.1 Background and motivation

During the last decades there have been a vast expansion of computer-based control of vehicles. Mostly due to the fact that the computer power rapidly increases and a computer becomes smaller and cheaper for every day that passes. The purpose of the controller can for example be to help the driver, increase comfort or to completely maneuver the vehicle. The last part is called autonomous driving and is what this work is going to explore. In everyday life autonomous driving becomes more and more common; a good example is the lawn mower. In a future perspective autonomous driving has an extreme potential and many vehicle companies allocate a lot of manpower for research and development of autonomous and semi-autonomous vehicles.

1.2 Problem formulation

The main purpose of this thesis is to investigate if a sampling-based motion planning algorithm called Closed-Loop Rapidly-exploring Random Tree (CL-RRT) can be used as a motion planner for a truck and trailer system for both backward and forward motion. For planning, a model of the system has to be proposed. The
Introduction

Direction of motion is critical for this application since the truck and trailer system has two rigid free joints and is therefore unstable in backward motion [5], [6]. To fit into the CL-RRT framework a closed-loop stable system has to be designed with a stabilizing controller. The system will be evaluated both in simulations and on the small-scale truck and trailer in Figure 1.1 using LEGO NXT.

![The small-scale truck and trailer used to evaluate performance of the motion planner. The system is built with LEGO NXT.](image)

1.3 Related work

The CL-RRT algorithm was originally developed by a research team at MIT for the 2007 DARPA Urban Challenge [8]. Later this algorithm was further developed and implemented for the iQMatic project, [11]. The CL-RRT algorithm has mostly been used as a real-time motion planner on cars, where it has shown promising results, see e.g. [10], [22], [14] and [8]. In [8] they use a path following controller called Pure pursuit as a low-level controller. This controller stabilizes the car to make the midpoint of the rear axle follow an reference path.

The RRT algorithm finds a suboptimal feasible solution to a non-convex optimal control problem, also known as a motion planning problem. The approach is from the beginning based on the RRT [18], where a tree of kinodynamically feasible trajectories is found by randomly generate sampling points and then perform forward simulation of the open-loop system. With the extensions proposed in [8] where they instead sample in the input space to the controller and then performing forward simulations of the closed-loop system, this planning algorithm can be used on an open-loop unstable system that is controlled such that the closed-loop system is stable. There are numerous of other advantages stated in [8] with the CL-RRT compared to the standard RRT. First, the prediction error of the forward simulations for the closed-loop system will be much lower than
the prediction error of the forward simulations for the open-loop system. Second, robustness against model errors and other disturbances will be better. Third, because the controller handles low-level tracking, the CL-RRT can focus on planning on a macro level by giving the controller straight-lines as input. This strategy simplifies the tree expansion and is suitable for real-time motion planning. In [22] they conclude that the CL-RRT algorithm is general and it can certainly be used as a motion planner for many types of systems.

A recursive nonlinear model for the truck and trailer systems based on kinematic constraints, was presented in [6]. In [27],[15], [25], [24], they design feedback controllers based on exact linearization for one and two axle truck and trailer systems with no kingpin hitching. In [6] they conclude that the general 3-trailer, due to the kingpin hitching, is not differentially flat and therefore feedback controllers based on exact linearization is not possible [27], [23]. In [6] they design a Linear Quadratic (LQ) controller to stabilize the system in backward and forward motion.

1.4 Outline

This thesis is organized in chapters with the following contents:
- Chapter 2 concerns a number of subjects to introduce the reader to the background theory of this thesis.
- Chapter 3 contains a description of the system platform used in this thesis; The small-scale truck and trailer, measurement systems, computers and other equipments.
- Chapter 4 describes the work to derive a suitable nonlinear model for this truck and trailer system and what characterizes it. The chapter also covers an analysis of different types of system constraints and model simplifications. A linearized version of the nonlinear model is also derived.
- Chapter 5 presents the decentralized low-level controller used for stabilizing and path-following of the truck and trailer system.
- Chapter 6 describes the CL-RRT motion planning algorithm used in this thesis.
- Chapter 7 presents the result of the controller and the motion planner both in a simulation environment and on the small-scale truck and trailer.
- Chapter 8 summarizes the main contributions of this thesis and discusses future work in this area of control theory.
This chapter is intended to provide necessary basic knowledge about different subjects covered in this thesis.

2.1 Linearization of a nonlinear model

A time-invariant nonlinear system with no static input-output relationship can be written on the form (2.1) where \( \mathbf{p} \) denotes the states of the system, \( \mathbf{u} \) denotes the control inputs and \( \mathbf{y} \) denotes the measurements of the states.

\[
\begin{align*}
\dot{\mathbf{p}} &= F(\mathbf{p}, \mathbf{u}) = A(\mathbf{p}) + B(\mathbf{p}, \mathbf{u}) \\
y &= H(\mathbf{p})
\end{align*}
\]  

(2.1a)  

(2.1b)

When using linear control theory, (2.1) has to be linearized around a stationary point \((\mathbf{p}_e, \mathbf{u}_e)\), where \( F(\mathbf{p}_e, \mathbf{u}_e) = 0 \). This can be done if \( F \) is one time continuously differentiable in a region of \((\mathbf{p}_e, \mathbf{u}_e)\). A first order taylor expansion of (2.1) yields,

\[
\begin{align*}
\dot{\mathbf{p}} &= A(\mathbf{p} - \mathbf{p}_e) + B(\mathbf{u} - \mathbf{u}_e) \\
y &= C\mathbf{p}
\end{align*}
\]

(2.2a)  

(2.2b)
where the matrices $A$, $B$ and $C$ are defined as

$$A = \left. \frac{\partial A(p)}{\partial p} \right|_{(p_e)} + \left. \frac{\partial B(p, u)}{\partial p} \right|_{(p_e, u_e)} \tag{2.3a}$$

$$B = \left. \frac{\partial B(p, u)}{\partial u} \right|_{(p_e, u_e)} \tag{2.3b}$$

$$C = \left. \frac{\partial H(p)}{\partial p} \right|_{(p_e)} \tag{2.3c}$$

The final linearized model, with $\bar{p} = p - p_e$ and $\bar{u} = u - u_e$ is

$$\dot{\bar{p}} = A\bar{p} + B\bar{u} \tag{2.4a}$$
$$y = C\bar{p} \tag{2.4b}$$

### 2.2 Linear Quadratic control

Given a linearized system (2.4) that is controllable, a Linear Quadratic controller (LQ) can locally stabilize an open-loop unstable system and control it to follow a specified reference signal, see [19]. The LQ controller can be written as

$$u = L_r(r) - L(p - p_e) \tag{2.5}$$

where $r$ is the reference signal. The feedback gain $L$ is designed by solving the optimal control problem

$$\min_u J = \min_u \int_0^\infty (\dot{\bar{p}}^T Q_1 \dot{\bar{p}} + \bar{u}^T Q_2 \bar{u}) dt \tag{2.6}$$

together with (2.4). The solution depends on the positive semi-definite weight matrices $Q_1$ and $Q_2$. The matrices $Q_1$ and $Q_2$ are often chosen to be diagonal. The optimal feedback gain becomes

$$L = Q_2^{-1} B^T P \tag{2.7}$$

where $P$ is the solution to the algebraic Riccati equation (2.8), where $A$ and $B$ are defined by (2.4).

$$Q_1 + A^T P + PA - PBQ_2^{-1} B^T P = 0 \tag{2.8}$$
Generally $L_r$ can be written as $L_r(r) = u_d$, where $u_d$ is the desired control signal. The LQ controller becomes

$$u = u_d - L\dot{p}$$  \hfill (2.9)$$

In the special case with only one control signal $u$, $Q_2$ is a scalar and the cost function (2.6) can be manipulated in the following way with $Q \equiv Q_1/Q_2$

$$\min_u J = \min_u Q_2 \int_0^\infty (\dot{p}^T Q \dot{p} + \ddot{u}^2)dt = \min_u \int_0^\infty (\dot{p}^T Q \dot{p} + \ddot{u}^2)dt$$ \hfill (2.10)$$

With the solution

$$0 = Q + A^T P + PA - PBB^T P$$ \hfill (2.11a)$$

$$L = B^T P$$ \hfill (2.11b)$$

$$u = u_d - L\dot{p}$$ \hfill (2.11c)$$

### 2.3 Constrained motion

Constrained motion can be separated into two groups of constraints: **Holonomic constraints** and **Nonholonomic constraints**. What is common with these constraints is that they lower the **degrees of freedom** for a system with the same amount as the number of linear independent constrained equations. These constraints are often related to vehicles and other robotic systems. The constraints will in these cases give restrictions in the velocity vector of the vehicle or describe the rigidity of the system, i.e. a car can not move sideways. The following definition is based on [26]. Given a state-space representation of a system with $n$ states, if a given set of $k < n$ relationships between the states can be formed by smooth functions

$$h_i(q) = 0, \quad i = 1, ..., k,$$ \hfill (2.12)$$

they are called holonomic constraints. The motion of the system then lies on an $m = n - k$-dimensional hypersurface (integral manifold) defined by $h_i(q(t))$, $\forall t > 0$. A set of $p < n$ constraints on the velocity

$$< w_j(q), \dot{q} > = 0, \quad j = 1, ..., p,$$ \hfill (2.13)$$

is called **Pfaffian constraints**, [18]. If (2.13) is integrable then the pfaffian constraints are also holonomic otherwise they are said to be nonholonomic.

A kinematic model of a car, with $p = [X, Y, \theta]^T$, (see Figure 2.1) is said to be
nonholonomic and has two degrees of freedom since it rules under the pfaffian constraint

\[ \dot{X} \sin \theta - \dot{Y} \cos \theta = 0 \]  

(2.14)

because \( v_{ort} \) (see Figure 2.1) has to be zero. The general solution of (2.14) is

\[ \dot{X} = v \cos \theta \]  

(2.15a)

\[ \dot{Y} = v \sin \theta \]  

(2.15b)

During a short distance segment \( ds \) the steering angle can be assumed fixed, we get \( ds = Rd\theta \). By using trigonometry from Figure 2.1 we get \( R = \frac{L}{\tan \alpha} \) and hence

\[ \dot{\theta} = \frac{v}{L} \tan \alpha. \]  

(2.16)

Relationship (2.15) and (2.16) together defines a third order system with two degrees of freedom. Hereafter we will refer to (2.15)-(2.16) as a kinematic model of a car.

\textbf{Figure 2.1:} A kinematic model of a car with the generalized coordinates \( p = [X, Y, \theta]^T \) and velocity \( v \) defined from the rear axle.
The truck and trailer system is a general 2-trailer [6] with four axles, two rigid free joints, a kingpin hitching and an actuated front steering. A small-scale truck and trailer is built with Lego for testing and evaluation of performance. LEGO-NXT has been used as a hardware and software platform because it provides all the basic motors and sensors that is needed to reproduce a real truck and trailer, see Figure 1.1. The LEGO-NXT platform consists of two servo motors [3], two angular sensors [1] and a programmable brick [2]. Some geometric lengths had to be chosen which are defined in Figure 4.1. For our system we chose, length of the truck $L_1 = 19$ cm, length of the kingpin hitching $M_1 = 3.6$ cm, length of the dolly $L_2 = 14$ cm and length of the trailer $L_3 = 33$ cm. With these geometric length the scale is of around 1:30 of a commercial truck and trailer system. A quite powerful PC with a Quad Core Xeon processor is used to perform motion planning. Communication between the LEGO-NXT and the PC was done via Bluetooth.

### 3.1 Servo motors

To generalize a truck and trailer system two servo motors [3] is needed. One for the velocity on the rear wheels of the truck and one for the front wheel steering. The servo motors is controlled with an integrated PID-controller. There were two different ways to set the reference to the servo motor, either you set a specific angle or you set a specific angular speed. For the servo motor that was generating the velocity the angular speed was set and for the steering mechanism instead a specific angle was set.
3.2 Steering mechanism

The steering mechanism consists of a servo motor which is connected to the steering wheels via a transformation consisting of gearwheels. Due to the gearwheels a dead zone is introduced when changing sign on the steering angle rate. The backlash was measured to approximately 6 degrees. Based on Figure 3.1 the servo motor is directly, via a scalar, changing $dl$. The relationship between the motor angle and steering angle becomes

$$\sin \alpha = \frac{dl}{r} \Rightarrow \alpha_m = C_0 \sin \alpha$$

where $\alpha_m$ is the motor angle, $\alpha$ is the steering angle and $C_0 = 320$ is a constant that we have identified by measure the total steering angle region together with corresponding motor angle region $\alpha_m$. The maximum steering angle was design to be 45 degrees.

![Figure 3.1: The relationship between the steering angle and the motor angle for the Lego platform.](image)

3.3 Measurement system

To be able to plan and control the Lego platform measurements of the relative angles of the rigid free joints and global position and orientation of the system is needed, see Figure 4.1.

3.3.1 Angular sensors

The relative angles $\beta_2$ and $\beta_3$ are crucial to measure. To achieve good control performance accurate measurements of these angles is preferred. This can be provided by LEGO-NXT angular sensors [1]. The sensors directly provide absolute angle $0 \leq \phi \leq 2\pi$ and an accumulated angle to a certainty of $0.0175$ rad, (1 deg) [1]. With help of a gearbox consisting of two gearwheels this accuracy can be increased even more. With a gear ratio $N = 10$ the accuracy was to approximately $\frac{0.0175}{N}$ rad. But due to weakness in the connection from the gearing to the angular sensor the accuracy is not really that good.
3.3.2 Position system using IR-cameras

With a measurement system consisting of two IR-cameras we are able to measure global orientation and position of the small-scale truck and trailer. The measurement system is using two IR-cameras from the roof which detects IR-markers on the vehicle. A project group at LIU, partly consisting of the author, integrated the small-scale truck and trailer in this measurement system. How the communication between computers works can be found in [4]. The global orientation and the position is filtered by a Kalman filter where a constant velocity model is used as reference model. A measurement system is never flawless and will introduce uncertainties in the measured states of the system. An analysis of performance of the measurement system can be found in [4].
In this chapter a nonlinear model of a truck and trailer system is derived based on holonomic and nonholonomic paffian constraints. The constraints arise from the assumption of rolling without slipping of the wheels. The states, or the generalized coordinates, is $p = [x_3, y_3, \theta_3, \beta_3, \beta_2]^T$ which correspond to the flat outputs, [5], [27], [25], [23]. For future control design a linearized version is derived and physical input and state constraints for the system is presented. Also model simplifications are discussed.

Figure 4.1: A schematic picture of the truck and trailer system with a definition of the generalized coordinates for the system. Note that all angles are set positive counter clockwise.
### 4.1 Recursive formula based on kinematic constraints

Assuming $\dot{\theta}_i$ and $v_i$ are known we need to derive a recursive formula that describes the velocity $v_{i+1}$ and the angular rate $\dot{\theta}_{i+1}$ in terms of $\dot{\theta}_i$, $v_i$ and known geometric lengths, see Figure 4.2. This has been done for a general N-trailer in [5], [6]. For the sake of completeness these calibrations have also been represented here. The assumption of rolling without slipping of the wheels can be formulated in terms of nonholonomic pfaffian constraints. The velocity of each body in the direction perpendicular to its rear wheels must be zero. Using this on the rear axle of the $i$-th and $i+1$-th trailer (see Figure 4.2) we get

$$
\begin{align*}
\dot{x}_i \sin \theta_i - \dot{y}_i \cos \theta_i &= 0 \\
\dot{x}_{i+1} \sin \theta_{i+1} - \dot{y}_{i+1} \cos \theta_{i+1} &= 0 
\end{align*}
$$

(4.1)

which yields the solutions

$$
\begin{align*}
\dot{x}_i &= v_i \cos \theta_i \\
\dot{y}_i &= v_i \sin \theta_i \\
\dot{x}_{i+1} &= v_{i+1} \cos \theta_{i+1} \\
\dot{y}_{i+1} &= v_{i+1} \sin \theta_{i+1}
\end{align*}
$$

(4.2)

If $L_{i+1}$ is the distance between the $i+1$-th axle and the hitching point of the $i$-th trailer and $M_i$ is the distance between the $i$-th trailer axle and same hitching
point we get the following holonomic relationships

\[
\begin{align*}
x_i - x_{i+1} &= L_{i+1} \cos \theta_{i+1} + M_i \cos \theta_i \\
y_i - y_{i+1} &= L_{i+1} \sin \theta_{i+1} + M_i \sin \theta_i
\end{align*}
\] (4.3)

These constraints have to hold for all time instances and therefore also its corresponding derivative constraints

\[
\begin{align*}
\dot{x}_{i+1} - \dot{x}_i &= L_{i+1} \sin \theta_{i+1} \dot{\theta}_{i+1} + M_i \sin \theta_i \dot{\theta}_i \\
\dot{y}_{i+1} - \dot{y}_i &= L_{i+1} \cos \theta_{i+1} \dot{\theta}_{i+1} + M_i \cos \theta_i \dot{\theta}_i
\end{align*}
\] (4.4)

Inserting (4.2) into (4.4) then yields

\[
\begin{align*}
v_{i+1} \cos \theta_{i+1} - v_i \cos \theta_i &= L_{i+1} \sin \theta_{i+1} \dot{\theta}_{i+1} + M_i \sin \theta_i \dot{\theta}_i \hspace{1cm} (4.5a) \\
v_i \sin \theta_i - v_{i+1} \sin \theta_{i+1} &= L_{i+1} \cos \theta_{i+1} \dot{\theta}_{i+1} + M_i \cos \theta_i \dot{\theta}_i \hspace{1cm} (4.5b)
\end{align*}
\]

By eliminating \(v_{i+1}\) from (4.5) we get the desired recursive equation for the angular rate

\[
\dot{\theta}_{i+1} = \frac{v_i \sin (\theta_i - \theta_{i+1})}{L_{i+1}} - \frac{M_i \cos (\theta_i - \theta_{i+1}) \dot{\theta}_i}{L_{i+1}} \hspace{1cm} (4.6)
\]

If we instead eliminate \(\dot{\theta}_{i+1}\) from (4.5) we get the desired recursive equation for the velocity

\[
v_{i+1} = v_i \cos (\theta_i - \theta_{i+1}) + M_i \sin (\theta_i - \theta_{i+1}) \dot{\theta}_i \hspace{1cm} (4.7)
\]

Finally the generalized angle rate becomes

\[
\dot{\beta}_{i+1} = \dot{\theta}_i - \dot{\theta}_{i+1} \hspace{1cm} (4.8)
\]

Since

\[
\beta_{i+1} = \theta_i - \theta_{i+1} \hspace{1cm} (4.9)
\]

### 4.2 Derivation of nonlinear model for a general 2-trailer system

A schematic picture of the truck and trailer system is shown in Figure 4.1. The states are \(p = [x_3, y_3, \theta_3, \beta_3, \beta_2]^T\), where \((x_3, y_3)\) denotes the center of the rear axle position in global coordinates, \(\theta_3\) is the absolute orientation of the trailer, \(\beta_3\) is the orientation of the dolly with respect to the trailer, \(\beta_2\) is the orientation of the truck with respect to the dolly and \(\alpha\) is the steering angle. The geometric lengths of each separate body are also defined in Figure 4.1. The speed \(v\) is the longitudinal velocity of the truck’s rear axle. The speed \(v\) and the steering angle \(\alpha\) is considered as control inputs to the system, \(u = [\alpha, v]^T\). A kinematic model for this truck and trailer system can be derived using the recursive equations (4.6) - (4.8), [6].

Starting from the truck (see Figure 4.1) we get \(v_1 \equiv v\) and by using the fact that
the truck has the same kinematic description as a kinematic model of a car (2.15), (2.16) we get \( \dot{\theta}_1 = \frac{v}{L_1} \tan \alpha \). Further, by using the recursive equations (4.6) - (4.8), we get:

\[
\begin{align*}
\dot{\theta}_2 &= \frac{v \sin (\theta_1 - \theta_2)}{L_2} - \frac{M_1 \cos (\theta_1 - \theta_2) \dot{\theta}_1}{L_2} = v \left( \frac{\sin \beta_2}{L_2} - \frac{M_1}{L_1 L_2} \cos \beta_2 \tan \alpha \right) \quad (4.10a) \\
v_2 &= v \cos (\theta_1 - \theta_2) + M_1 \sin (\theta_1 - \theta_2) \dot{\theta}_1 = v \cos \beta_2 \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \quad (4.10b) \\
\dot{\beta}_2 &= \dot{\theta}_1 - \dot{\theta}_2 = v \left( \frac{\tan \alpha}{L_1} - \frac{\sin \beta_2}{L_2} + \frac{M_1}{L_1 L_2} \cos \beta_2 \tan \alpha \right) \quad (4.10c)
\end{align*}
\]

By using the same recursive equations again we get (with \( M_2 = 0 \)):

\[
\begin{align*}
\dot{\theta}_3 &= \frac{v_2 \sin (\theta_2 - \theta_3)}{L_3} = \frac{v \sin \beta_3 \cos \beta_2}{L_3} \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \quad (4.11a) \\
v_3 &= v_2 \cos (\theta_2 - \theta_3) = v \cos \beta_3 \cos \beta_2 \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \quad (4.11b) \\
\dot{\beta}_3 &= \dot{\theta}_2 - \dot{\theta}_3 = v \cos \beta_2 \left( \frac{1}{L_2} \left( \tan \beta_2 - \frac{M_1}{L_1} \tan \alpha \right) - \frac{\sin \beta_3}{L_3} \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \right) \quad (4.11c)
\end{align*}
\]

The derivation of \( \dot{x}_3 \) and \( \dot{y}_3 \) is now straightforward since

\[
\begin{align*}
\dot{x}_3 &= v_3 \cos \theta_3 = v \cos \beta_3 \cos \beta_2 \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \cos \theta_3 \quad (4.12a) \\
\dot{y}_3 &= v_3 \sin \theta_3 = v \cos \beta_3 \cos \beta_2 \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \sin \theta_3 \quad (4.12b)
\end{align*}
\]

The final nonlinear model becomes

\[
\begin{align*}
\dot{x}_3 &= v \cos \beta_3 \cos \beta_2 \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \cos \theta_3 \quad (4.13a) \\
\dot{y}_3 &= v \cos \beta_3 \cos \beta_2 \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \sin \theta_3 \quad (4.13b) \\
\dot{\theta}_3 &= \frac{v \sin \beta_3 \cos \beta_2}{L_3} \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \quad (4.13c) \\
\dot{\beta}_3 &= v \cos \beta_2 \left( \frac{1}{L_2} \left( \tan \beta_2 - \frac{M_1}{L_1} \tan \alpha \right) - \frac{\sin \beta_3}{L_3} \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \right) \quad (4.13d) \\
\dot{\beta}_2 &= v \left( \frac{\tan \alpha}{L_1} - \frac{\sin \beta_2}{L_2} + \frac{M_1}{L_1 L_2} \cos \beta_2 \tan \alpha \right) \quad (4.13e)
\end{align*}
\]

It has been shown in [5], [6] that the off hitching \( (M_1 \neq 0) \) implies that the system is neither differentially flat nor feedback linearizable. The nonlinear model holds under the condition of rolling without slipping. Since this thesis is about low speed maneuvers this assumption will not be far from true. The sign of \( v \) decides
the direction of motion where \( v < 0 \) implies backward motion and \( v > 0 \) implies forward motion. Notice that the speed \( v \) only enters linearly in (4.13) and the model can be described on the following form

\[
\dot{p} = v(A(p) + B(p, \alpha)) \tag{4.14}
\]

Introduce the arc length \( ds \equiv |v|dt \) and equation (4.14) can be rewritten as (4.15).

\[
\frac{dp}{ds} = \frac{v}{|v|}(A(p) + B(p, \alpha)) \tag{4.15}
\]

From this expression we see that it is only the sign of \( v \) that is interesting. Therefore it is motivated to decouple the speed. To fit into the framework of the path-planner it is interesting to stabilize the states \( \beta_2 \) and \( \beta_3 \) with a low level feedback controller. This is because of the fact that it is these states, with their corresponding rigid free joints, that are introducing the unstable behavior in backward motion. A high level controller is then assumed to control the rest of the states and the speed \( v \). Therefore consider the subsystem (4.16) of (4.13) with the corresponding states \( p = [\beta_3, \beta_2]^T \).

\[
\dot{\beta}_3 = v \cos \beta_2 \left( \frac{1}{L_2} \tan \beta_2 - \frac{M_1}{L_1} \tan \alpha \right) - \frac{\sin \beta_3}{L_3} \left( 1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \tag{4.16a}
\]
\[
\dot{\beta}_2 = v \left( \tan \alpha \frac{1}{L_1} - \frac{\sin \beta_2}{L_2} + \frac{M_1}{L_1 L_2} \cos \beta_2 \tan \alpha \right) \tag{4.16b}
\]

### 4.3 Linearization along arc circles

For this application it is critical to stabilize the states \( p = [\beta_3, \beta_2]^T \). To close the loop a LQ controller will be used and hence we need to derive a linear state-space model of the subsystem (4.16). A general linearization of a nonlinear system around an equilibrium point \( (p_e, u_e) \) yields

\[
\dot{p} = v \left( \frac{\partial A(p)}{\partial p}(p_e) + \frac{\partial B(p, u)}{\partial p}(p_e, u_e) \right)(p - p_e) + v \frac{\partial B(p, u)}{\partial u}(p_e, u_e)(u - u_e) \tag{4.17}
\]

To fit into the path-planning framework we linearize the nonlinear model around arc circles with different radius. This can be performed due to the fact that a constant steering angel \( \alpha_e \) in stationarity will correspond to an equilibrium value for both \( \beta_{2e} \) and \( \beta_{3e} \) see Figure 4.3. From Figure 4.3, using a constant steering angle \( \alpha_e \) we can derive the equilibrium configuration (4.18) based on geometry

\[
\beta_{3e} = \text{sign}(\alpha_e) \arctan \left( \frac{L_3}{R_3} \right), \quad \beta_{2e} = \text{sign}(\alpha_e) \left( \arctan \left( \frac{M_1}{R_1} \right) + \arctan \left( \frac{L_2}{R_2} \right) \right) \tag{4.18}
\]

Where \( R_1 = L_1/\tan \alpha_e, \ R_2 = \sqrt{R_1^2 + M_1^2 - L_2^2} \) and \( R_3 = \sqrt{R_2^2 - L_3^2} \) are obtained from the Pythagorean theorem and are the radius of each axle's corresponding circular trajectory. Inserting the equilibrium configuration in (4.16) yields, \( \dot{p} = 0 \).
Figure 4.3: A schematic picture of the truck and trailer system with a circular stationary state configuration.

From Figure 4.3 we conclude that the steering angle $\alpha_e$ will have the same sign as $\beta_{2,e}$ and $\beta_{3,e}$ due to their definitions from Figure 4.1. Linearizing the system (4.18) around its equilibrium point $(p_e, \alpha_e)$ then yields,

$$\dot{p} = v(\bar{A}(p - p_e) + \bar{B}(\alpha - \alpha_e))$$ (4.19)

where

$$\bar{A} = \begin{pmatrix}
\cos \beta_{2,e} \cos \beta_{3,e} & \frac{M_1}{L_2} \cos \beta_{3,e} \sin \beta_{2,e} \tan \alpha_e \\
\frac{M_1}{L_1} \cos \beta_{3,e} + \frac{M_1}{L_2} \sin \beta_{2,e} \sin \beta_{3,e} & \frac{M_1}{L_2} \left( \cos \beta_{2,e} + \frac{M_1}{L_1} \sin \beta_{2,e} \tan \alpha_e \right) + \frac{M_1}{L_1} \left( \cos \beta_{3,e} \sin \beta_{2,e} \tan \alpha_e \right)
\end{pmatrix}$$ (4.20)

$$\bar{B} = \begin{pmatrix}
\frac{M_1}{L_2} \left( \cos \beta_{2,e} \frac{\sin \beta_{2,e} \sin \beta_{3,e}}{L_1} \tan \alpha_e \right) (1 + \tan^2 \alpha_e) \\
\frac{M_1}{L_1} \left( \cos \beta_{2,e} \frac{\sin \beta_{2,e} \sin \beta_{3,e}}{L_1} \tan \alpha_e \right) (1 + \tan^2 \alpha_e)
\end{pmatrix}$$ (4.21)

There exist limits for the states $\beta_{2}$ and $\beta_{3}$ when the system reaches their physical limits. At these limits the trailer is in a configuration called a jack knife and it’s no longer possible for the trailer to go further backwards.

4.4 Modeling steering angle dynamics in simulations

A significant model simplification is that we have chosen not to model steering angle dynamics in the nonlinear model due to the fact that the model is already complex as it is. But this is an important subject and the steering angle dynamics will be modeled separately when doing simulations and design of the controllers. The steering angle reaches saturation in $\alpha_s$ which can be modeled as

$$|\alpha| \leq \alpha_s = \pi/4 \text{ rad} = 45 \text{ deg.}$$ (4.22)

From experiments on our test platform we found the rate limit of the steering angle to be 180 deg/s. This constraint has been modeled as a first order system with a time constant, $T = 1 \text{ s}$. One big problem with the steering mechanism
4.4 Modeling steering angle dynamics in simulations

on the test platform is that it has backlash. This backlash is introduced by the angular motor via the transformation to steering angle, which consists of gear wheels. The dead zone was measured to approximately 0.1 rad. One other problem with the steering mechanism is that it has to be set manually to zero before start. This calibration can not be done perfectly and therefore we will also have a bias in the steering angle $\alpha_{bias} \approx 0.1$ rad. The steering angle dynamics modeled in Simulink can be seen in Figure 4.4. Since both the backlash and the steering angle saturation is nonlinear dynamics the order where a block is placed does matter.

Figure 4.4: The steering angle dynamics modeled in Simulink with saturation, time constant, backlash and bias.
Design of closed-loop system

The CL-RRT algorithm is intended to be used on a closed-loop stable system. Because the truck and trailer is unstable in backward motion, a stabilizing controller has to be designed. A Pure Pursuit controller [8], [9] has already been implemented [11] and because the truck and trailer system, in forward motion, can be seen as a kinematic model of a car the Pure pursuit controller was also used in this thesis. In fact, it will further down in this chapter be shown that the pure pursuit controller can be used to control the truck and trailer in backward motion as well. For stabilization of the two rigid free joints a gain scheduled LQ controller will be designed based on [6]. The resulting closed-loop system will therefore be controlled by a hybrid nonlinear controller with a switching scheme depending on the direction of motion. A schematic picture of the hybrid closed-loop system can be seen in Figure 5.1a and 5.1b.

(a) A block diagram of the closed-loop system in forward motion.

(b) A block diagram of the closed-loop system in backward motion.

Figure 5.1: The hybrid control scheme. The direction of motion is based on the sign of \( v \).
5.1 Derivation of the pure pursuit control law

A Pure pursuit controller is used for steering a vehicle to follow a specific path; in this thesis it will be paths in the two dimensional space. This path following controller was chosen because it has shown nice performance on both aerial and ground vehicle applications [16], [8]. Here follows the derivation of the pure pursuit controller based on [9], [7]. The concept of the pure pursuit controller is that it tries to pursue a look-ahead point, in Figure 5.2 referred to as $P$. This point is given by the intersection between a reference path and a look-ahead circle. The look-ahead circle has the center in the reference position of the car $P^*$ and a radius $R$. Given this look-ahead point the controller sets a desired control signal such that the vehicle will end up in this point if the control signal is hold constant. But for each time step the intersection between the reference path and the look-ahead circle will move together with the vehicle. This will result in a smooth path if the look-ahead distance is chosen big enough. Using the definitions in Figure 5.2, where $P^* = [X, Y]^T_{P^*}$ and $\theta$ is the global position and orientation of the vehicle, $L$ is the length of the vehicle, $r$ is the curvature radius, $R$ is the look-ahead distance, $\theta_e$ is the error heading, $P = [x, y]^T_c$ is the coordinates of the look-ahead point $P$ in the local frame of the car. We get the following relationships from geometry:

\begin{align*}
  x + d &= r \\
  x^2 + y^2 &= R^2 \\
  y^2 + d^2 &= y^2 + (r - x)^2 = r^2 \Rightarrow \\
  r^2 - 2rx + x^2 + y^2 &= r^2 \Rightarrow r = \frac{R^2}{2x} = \frac{R^2}{2R \sin \theta_e} = \frac{R}{2 \sin \theta_e}
\end{align*}
By proposing that the vehicle is well described by a kinematic car (2.15) - (2.16), with the same definitions as in Figure 5.2 the motion model is

\[
\begin{align*}
\dot{X} &= v \cos \theta \\
\dot{Y} &= v \sin \theta \\
\dot{\theta} &= \frac{v}{L} \tan \alpha
\end{align*}
\] (5.2)

If we assume a coordinated turn with a constant curvature radius \( r \) and constant velocity \( v \) of the rear axle of the car, we get \( v = -r \dot{\theta} \) (negative sign because \( \theta \) is defined positive counter clockwise). By using (5.2) and (5.1d), the control law for the steering angle \( \alpha \) is

\[
\alpha = -\arctan \left( \frac{L}{r} \right) = -\arctan \left( \frac{2L \sin \theta_e}{R} \right)
\] (5.3)

Finally with use of \( \arctan_2 \) which is a generalization of \( \arctan \) in the region \(-\pi \leq \phi \leq \pi\) we get

\[
\theta_e = \arctan_2(X_p - X, Y_p - Y) - \theta
\] (5.4)

where \( P = [X_p, Y_p]^T \) is the look ahead point. The look ahead distance \( R \) is a design parameter for the pure pursuit controller and has to be chosen big enough to make the closed system stable. A big \( R \) will make the path smooth and the car will take short cuts. This will decrease the maneuverability of the car. Naturally if we use the same control law (5.3) in backward motion it will take the car back from \( P \) to \( P^* \). Note that the pure pursuit control law holds for all \( \theta_e \) but is only working as intended in the region \(-\pi/2 \leq \theta_e \leq \pi/2\).

### 5.2 Closed-loop system in forward motion

In forward motion the reference position is moved to \([x_1, y_1, \theta_1]^T\) which is the midpoint of the rear axle of the truck and its global orientation, see Figure 5.3. The Pure pursuit control law becomes

\[
\begin{align*}
\dot{X} &= v_1 \cos \theta_1 \\
\dot{Y} &= v_1 \sin \theta_1 \\
\dot{\theta}_1 &= \frac{v_1}{L_1} \tan \alpha
\end{align*}
\]

\( L_1 \) is the length of the truck. The Pure pursuit control law in forward motion is given by

\[
\alpha = -\arctan \left( \frac{L_1}{r_1} \right) = -\arctan \left( \frac{2L_1 \sin \theta_{e1}}{R_1} \right)
\] (5.5)

Finally with use of \( \arctan_2 \) which is a generalization of \( \arctan \) in the region \(-\pi \leq \phi \leq \pi\) we get

\[
\theta_{e1} = \arctan_2(X_p - X, Y_p - Y) - \theta_1
\] (5.6)

where \( P = [X_p, Y_p]^T \) is the look ahead point. The look ahead distance \( R_1 \) is a design parameter for the pure pursuit controller and has to be chosen big enough to make the closed system stable. A big \( R_1 \) will make the path smooth and the car will take short cuts. This will decrease the maneuverability of the car. Naturally if we use the same control law (5.5) in backward motion it will take the car back from \( P \) to \( P^* \). Note that the pure pursuit control law holds for all \( \theta_{e1} \) but is only working as intended in the region \(-\pi/2 \leq \theta_{e1} \leq \pi/2\).

**Figure 5.3:** A presentation of the geometry and states used in the control law for the pure pursuit controller in forward motion.
Design of closed-loop system

\[
\alpha = -\arctan \left( \frac{2L_1 \sin \theta_e}{R} \right) 
\]  
(5.5)

with

\[
\theta_e = \arctan_2 (X_p - x_1, Y_p - y_1) - \theta_1
\]  
(5.6)

5.3 Closed-loop system in backward motion

In backward motion the reference position is moved to \([x_3, y_3, \theta_3]^T\) which corresponds to the flat outputs of the truck and trailer system, [6]. For reversing the truck and trailer system a feedback controller will be designed to make the closed-loop system locally stable. This can be achieved with a gain scheduled Linear Quadratic controller, [6], [13]. In backward motion a path following controller, based on Pure pursuit, will be used to give suitable reference values for the gain scheduled LQ controller.

![Diagram](image)

*Figure 5.4: A presentation of the geometry and states used in the control law for the pure pursuit controller in backward motion.*

5.3.1 Pure pursuit controller in backward motion

Using the fact that the trailer’s acting steering wheel is \(\beta_3\), the Pure pursuit controller can be used to generate suitable reference values for this angle, [21]. Given the global orientation \(\theta_3\) and the midpoint of the rear axle of the trailer \([x_3, y_3]^T\) as reference position for the pure pursuit controller, see Figure 5.4. The control law becomes

\[
\beta_{3,d} = -\arctan \left( \frac{2L_3 \sin \theta_e}{R} \right)
\]  
(5.7)

with

\[
\theta_e = \arctan_2 (X_p - x_3, Y_p - y_3) - \theta_3
\]  
(5.8)
The control law (5.4) is used to generate suitable reference values for the gain scheduled LQ controller. Due to complex dynamics from steering angle to $\beta_3$, there will exist a configuration delay between desired value and actual value. To compensate for this we will have a quite large look ahead distance $R$ and add a proportionality control on the desired angle. A large look ahead distance is chosen to give the LQ controller time to reconfigure the system and to keep the rate change in the reference angle $\beta_3$ small to maintain stability. The proportionality controller is then used to make the controller more aggressive and to compensate for configuration delay. The proposed extend Pure pursuit controller becomes

$$\beta_{3,d,temp} = -\arctan\left(\frac{2L_3 \sin \theta_e}{R}\right) \quad (5.9a)$$

$$\beta_{3,d} = \beta_{3,d,temp} + K_p(\beta_{3,d,temp} - \beta_3) \quad (5.9b)$$

Further in this chapter we will analyze the closed-loop system without this proportionality control, i.e. $K_p = 0$.

### 5.3.2 Feedback controller

The gain scheduled LQ controller should be able to locally stabilize the system around a desired reference signal. The reference signal is generated by a Pure pursuit controller, i.e. $r = \beta_{3,d}$. This is possible to achieve by a gain scheduled LQ controller with two degrees of freedom defined by (5.10).

$$u = L_r(\beta_{3,d}) - L(\mathbf{u}_d)(\mathbf{p} - \mathbf{p}_e(\mathbf{u}_d)) \quad (5.10)$$

The pre-compensation link, $L_r(\beta_{3,d})$, is a transformation from reference signal to desired input signal, e.g. $\mathbf{u}_d = L_r(\beta_{3,d})$. This input signal defines the working point of the feedback controller.

### 5.3.3 Gain scheduled feedback gain

To design our stabilizing feedback controller using LQ-theory we need a linearized system (4.19) and measurements of the states $\beta_2$ and $\beta_3$. To make the feedback controller work in a wide range of steering angles we need to derive a gain scheduled feedback controller. Given a desired steering angle $\alpha_d$, a specific feedback gain $L(\alpha_d)$ with a corresponding stationary point $\mathbf{p}_e = [\beta_{2,e}(\alpha_d) \; \beta_{3,e}(\alpha_d)]^T$ is chosen. Using the relationships described in (4.18) the states $\beta_{2,e}$ and $\beta_{3,e}$ are plotted in Figure 5.5a and 5.5b as a function of the steering angle.

Looking at Figure 5.5a one sees that $\beta_{3,e}$ is approaching $\pi/2$ rad when the steering angle is approaching 0.48 rad. This steering angle defines the maximum linearization point for a circular turn. Using the relations defined in 4.18 one sees that $\beta_{3,e} \to \pi/2$ is equivalent to $r_3 \to 0$. At this limit $r_2 = L_3$, combining this with the definition of $r_2$ we obtain

$$L_3^2 = L_1^2/\tan^2 \alpha_d + M_1^2 - L_2^2 \quad (5.11)$$
5 Design of closed-loop system

Equilibrium points for beta3

Equilibrium points for beta2

(a) The relationship between the state $\beta_3, e$ as a function of the steering angle $\alpha$ driving a circular turn.

(b) The relationship between the state $\beta_2, e$ as a function of the steering angle $\alpha$ driving a circular turn.

Figure 5.5: Two figures of the relationship between the equilibrium states as a function of the steering angle $\alpha$, in steady-state, driving a circular turn.

By extracting $\alpha_d$ from (5.11) and assuming that $0 < \alpha_d < \alpha_{\text{max}} < \pi/2$ we obtain

$$\alpha_{d,\text{max}} = \arctan \left( \sqrt{ \frac{L_1^2}{L_3^2 + L_2^2 - M_1^2} } \right)$$

(5.12)

Relationship (5.12) defines the limit for the linearization point of a truck and trailer system around a circular trajectory. By using the lengths for our specific truck and trailer system we get the limit $\alpha_{d,\text{max}} = 0.48$. By using the matrices $\bar{A}$ and $\bar{B}$ defined in (4.20), (4.21), the objective function $J$ is

$$J = \int_0^\infty (\bar{p}^T Q \bar{p} + \bar{\alpha}^2) \, dt$$

(5.13)

where $\bar{p} = p - p_e$ and $\bar{\alpha} = \alpha - \alpha_e$ is defined by (5.13). The feedback $\alpha_c$, becomes

$$\alpha_c = -L_c (\beta_2 e(\alpha_d), \beta_3 e(\alpha_d))$$

(5.14)

where $L_c$ is the solution to the algebraic Riccati equation (2.8). The feedback controller is then

$$\alpha = \alpha_d + \alpha_c = \alpha_d - L_{\beta_2}(\alpha_d)(\beta_2 - \beta_2 e(\alpha_d)) - L_{\beta_3}(\alpha_d)(\beta_3 - \beta_3 e(\alpha_d))$$

(5.15)

After testings in simulations the weight matrix was chosen to be $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. The gains are plotted in Figure 5.6a and 5.6b. The weight matrix $Q$ was chosen based under the following criterion (gathered from [19] and [20]):
5.3 Closed-loop system in backward motion

- How good is the signal to noise ratio of the system, a high value on the weight matrix will lead to a high feedback gain and hence noise will be magnified.
- The feedback gain has to be chosen big enough to stabilize the system locally and make this region of attraction as large as possible.
- To make the decentralized control strategy (Cascade control) work as intended the feedback controller has to be designed such that it is significantly faster than the Pure pursuit controller.

5.3.4 Nonlinear pre-compensation link in backward motion

Now we want to construct the pre-compensation link, \( L_r(\beta_{3,d}) \), that given a desired \( \beta_{3,d} \) a desired steering angle \( \alpha_d \) is obtained. Given an equilibrium point given by (4.19) we obtain

\[
\begin{align*}
    r_1 &= \frac{L_1}{\tan \alpha_d}, \\
    r_3 &= \frac{L_3}{\tan \beta_{3,d}}, \\
    r_1^2 &= L_3^2 + r_3^2 + L_2^2 - M_1^2
\end{align*}
\]  

(5.16)

By eliminating \( r_1 \) and \( r_3 \) from (5.16) the nonlinear transformation from \( \beta_{3,d} \) to \( \alpha_d \) becomes

\[
\alpha_d = \text{sign}(\beta_{3,d}) \arctan \left( \frac{L_1}{\sqrt{L_3^2 \left( 1 + \frac{1}{\tan^2 \beta_{3,d}} \right) + L_2^2 - M_1^2}} \right)
\]  

(5.17)

(5.17) is the searched expression for the pre-compensation link \( L_r(\beta_{3,d}) \).
5.4 Design and simulation results

Inserting the pure pursuit control law (5.7) in (5.17) we get the compact expression

\[
\alpha_d = \text{sign}(\beta_{3,d}) \arctan \left( \frac{L_1}{\sqrt{L_3^2 + \left( \frac{R}{2\sin \theta_e} \right)^2 + L_2^2 - M_1^2}} \right) \tag{5.18}
\]

The desired steering angle \( \alpha_d \) as function of the error heading \( \theta_e \) (5.18) is plotted in Figure 5.7 for some look ahead distances \( R \). From this figure it can be seen that a small look ahead distance is the same as an aggressive pre-compensation link.

![Pre-compensation link for different look ahead distance R](image)

**Figure 5.7:** The pre-compensation link is plotted with respect to error heading for different choice of look ahead distances \( R \).

To choose a well suited look ahead distance we first rely on simulations, then tune when testing on the real system. Since the test platforms steering mechanism has backlash \( (b = 0.05 \text{ rad}) \), a time constant \( (T_{\text{steer}} = 1 \text{ s}) \) and a possible calibration bias \( (\alpha_{\text{bias}} = 0.05 \text{ rad}) \) this was modeled when doing simulations to make the simulated system mimic the real system as good as possible. For evaluation of performance the velocity was set to \(-3 \text{ cm/s}\) and the closed-loop system was intended to follow a path consisting of three 90 degrees turns in a row. Three simulations with different look ahead distances \( (R = 66 \text{ cm}, R = 100 \text{ cm} \text{ and } R = 133 \text{ cm}) \) are presented in Figure 5.8, 5.9 and 5.10. From these plots it is hard to say which look ahead distance is most suited but with a trade of between good reference tracking against big steering angles, \( R = 100 \text{ cm} \) was selected as an initial guess. From these simulations we can conclude that the backlash and the bias in the steering mechanism introduce oscillations in the steering angle. It is interesting to analyze the impact of the backlash and the bias in the steering angle. To do so we have plotted the same path with \( R = 100 \text{ cm} \), where we have put the bias and the backlash to zero. The result can be seen in Figure 5.11. One can conclude that oscillations in the steering angle is gone as expected.
5.4 Design and simulation results

Figure 5.8: A simulation in Matlab with look ahead distance $R = 66$ cm and $v = -3$ cm/s. Both the motion of the truck and trailer (L.F) and the steering angle input together with the actual steering angle (R.F) are presented.

Figure 5.9: A simulation in Matlab with look ahead distance $R = 100$ cm and $v = -3$ cm/s. Both the motion of the truck and trailer (L.F) and the steering angle input together with the actual steering angle (R.F) are presented.

Compared to Figure 5.9 the overall trend in the steering angle is quite identical. One can also conclude that the paths don't differ that much and therefore we can say that the controller is not very sensitive to these types of model errors.
Simulation of truck and trailer system. (x₃(t), y₃(t))

**Figure 5.10:** A simulation in Matlab with look ahead distance $R = 133$ cm and $v = -3$ cm/s. Both the motion of the truck and trailer (L.F) and the steering angle input together with the actual steering angle (R.F) are presented.

Simulation of truck and trailer system. (x₃(t), y₃(t))

**Figure 5.11:** A simulation in Matlab with look ahead distance $R = 100$ cm and $v = -3$ cm/s. Both the motion of the truck and trailer (L.F) and the steering angle input together with the actual steering angle (R.F) are presented. In this simulation the backlash and the bias in the steering angle is set to zero.

### 5.5 Stability analysis of the closed-loop system

A stability analysis of the pure pursuit controller used to control a kinematic model of a car in forward motion has been done in [17]. These results can be directly transferred to this truck and trailer system in forward motion since our strategy is to use the midpoint of the truck’s rear axle as a reference position when driving forward. This means that the truck and trailer, in aspect of control,
is transformed to a kinematic model of a car in forward motion. In [6] they conclude that with a linear state feedback controller it is not possible to prove stability for the truck and trailer system in backward motion. They instead rely on an extensive numerical stability analysis by simulate the system from different initial states. In this thesis we have first of all a gain scheduled LQ controller which is a nonlinear controller and secondly we have a nonlinear pre-compensation link. Therefore we also have to rely on simulations.

As they also conclude in [6] it is the rigid free joints $\beta_2$ and $\beta_3$ that is by far the most critical states. Therefore it is interesting to do a numeric stability analysis by varying these initial states over a predefined grid and analyze if the system can recover to follow a straight line backwards. With this approach we will have an upper bound stability region for $\beta_2$ and $\beta_3$. With the same model of the steering angle dynamics (accept $\alpha_{bias} = 0$ rad to achieve symmetry), with the velocity set to $v = -1$ cm/s and with the look ahead distance $R = 100$ cm, two simulations can be seen in Figure 5.12 and 5.13 with different initial configurations. By doing simulations over a grid where $-0.9 \leq \beta_2 \leq 0.9$ and $-0.9 \leq \beta_3 \leq 0.9$ with an accuracy of 0.05 rad we got a stability region that is presented in Figure 5.14. It is very important to understand that this stability region is speed dependent. Redo the same stability analysis with $v = -3$ cm/s would result in a lower stability region since the system becomes more unstable in backward motion with increasing speed. Also different choices of parameters in the modeling of the steering angle dynamics will affect the stability region. From Figure 5.13 we can conclude that it is especially the maximum steering angle that affects the stability region since the steering angle in this case has a bang-bang characteristic. From the same figure we can conclude that the impact of backlash is speed dependent since the steering angle is not oscillating in this case as in Figure 5.10. Lower the speed is equivalent to lower the time constant of the steering angle. This means that if the steering mechanism is fast we can go faster backwards. From Figure 5.14 we conclude that the closed-loop system has a quite large stability region.

**Figure 5.12:** A simulation of the truck and trailer system from the initial states $x_3 = 0$, $y_3 = 0$, $\theta_3 = 0$, $\beta_3 = 0.35$ rad and $\beta_2 = -0.35$ rad. The look ahead distance was set to 100 cm. The closed-loop system is able to stabilize around the reference path.
This is a good result because it means that the CL-RRT algorithm will be able to test quite advanced maneuvers without losing stability.

Figure 5.13: A simulation of the truck and trailer system from the initial states $x_3 = 0$, $y_3 = 0$, $\theta_3 = 0$, $\beta_3 = \beta_2 = 0.68$ rad. The look ahead distance was set to 100 cm and $v = -1$ cm/s. The closed-loop system is able to stabilize around the reference path. Observe the bang-bang characteristics in the steering angle.

Figure 5.14: Simulated stability region for the closed-loop system following a straight line in backward motion from different initial states. The initial states $x_3$, $y_3$, $\theta_3$ are assumed to be zero. The look ahead distance was set to 100 cm and $v = -1$ cm/s. Yellow area marks jack-knife and blue area marks recovering to the straight line for the closed-loop system.
In this chapter we will explain the concept of the Closed-Loop Rapidly-exploring Random Tree (CL-RRT) algorithm, [8], [11], [12] used in this thesis. The CL-RRT algorithm will be used to find a suboptimal feasible solution to a non-convex optimal control problem (6.2) for the Lego platform described in Chapter 3.

6.1 The motion planning problem

Given a nonlinear state space description of a system

\[
\dot{p}(t) = f(p(t), u(t)), \quad p(0) = p_0 \tag{6.1}
\]

where \(p\) denotes the states of the system, \(u\) denotes the control inputs and \(p_0\) is the initial state configuration at time \(t = 0\). Assume there exists input constraints, \(u(t) \in U\) and constraints of the state configuration, \(p(t) \in X_{\text{free}}(t)\). Further more we demand that the final configuration \(p(t_f) \in X_{\text{goal}}\) holds. This can be formulated as an optimal control problem; calculate the control sequence \(u(t)\), \(t \in [0, t_f]\), \(t_f \in [0, \infty)\) that minimizes

\[
\begin{align*}
\min_u & \int_0^{t_f} \Gamma(X_{\text{free}}(t), p(t))dt + \Phi(p(t_f), X_{\text{goal}}) \\
\text{subject to} & \quad \dot{p}(t) = f(p(t), u(t)), \quad p(0) = p_0, \quad p(t_f) \in X_{\text{goal}} \\
& \quad p(t) \in X_{\text{free}}(t), \quad u(t) \in U
\end{align*} \tag{6.2}
\]

where \(\Gamma\) is a penalty of the configuration during the motion and \(\Phi\) is a penalty on final configuration. Generally this problem is hard to solve analytically and
numerical approaches have to be used. In this thesis we will explore the concept of the CL-RRT algorithm as an alternative solution strategy to (6.2).

6.2 Planner

The CL-RRT algorithm was originally developed by MIT for the 2007 DARPA Urban Challenge [8]. Later this algorithm was further developed and implemented by [11], [12]. The CL-RRT algorithm builds a tree of kinodynamically feasible trajectories by randomly explore the input space of the closed-loop stable system. The exploring part is called sampling. In this section the main parts needed for the algorithm will be presented.

6.2.1 Constraints on the final configuration

Given the motion planning problem (6.2) a goal region, $X_{\text{goal}}$, is used to define constraints on the final configuration of the states, $p(t_f)$. The goal region is constructed by first define $p_g$ as the optimal goal position, but all solutions in a close region of $p_g$ is also accepted as a final configuration. In this thesis we have used constraints that can be formulated as; given an goal position $p_g = [x_{3,g}, y_{3,g}, \theta_{3,g}, \beta_{3,g}, \beta_{2,g}]^T$ and an accepted deviation $\Delta = [\Delta r, \Delta \theta_3, \Delta \beta_3, \Delta \beta_2]^T$ the constraints are

$${\begin{cases} (x_3(t_f) - x_{3,g})^2 + (y_3 - y_{3,g})^2 \leq (\Delta r)^2 \\ |\theta_3(t_f) - \theta_{3,g}| \leq \Delta \theta_3 \\ |\beta_3(t_f) - \beta_{3,g}| \leq \Delta \beta_3 \\ |\beta_2(t_f) - \beta_{2,g}| \leq \Delta \beta_2 \end{cases}} \tag{6.3}$$

where we have used $\Delta r = 0.1$ m, $\Delta \theta_3 = 0.07$ rad, $\Delta \beta_3 = 0.08$ rad and $\Delta \beta_2 = 0.08$ rad. The size of this goal region has big influence on the computation time for the RRT algorithm, i.e. how many samples is needed before a feasible trajectory is found. We have chosen to have a quite large goal region and instead punish the deviation from $p_g$ in the objective function.

6.2.2 Simulation of closed-loop system

The hybrid controller designed in chapter 5 was implemented to be able to perform forward simulations of the closed-loop system. The nonlinear model (4.13) was forward simulated using an Euler forward approximation for the derivative (6.4), with the sampling time $T_s = 0.04$ s.

$$\dot{q} = F(q, u) \Rightarrow / \text{Euler forward} / \Rightarrow q[t + T_s] = q[t] + T_s F(q[t], u[t]) \tag{6.4}$$

The backlash and the bias in the steering mechanism was not modeled. The LQ controller was developed and designed offline. The look-ahead distances for the Pure pursuit controller was set to $R_{fwd} = 60$ cm and $R_{rev} = 100$ cm. The hybrid controller takes a reference signal consisting of an ordered list of triples
(X, Y, v_{cmd}) \text{; for } i \in \{1, 2, ..., n_{ref}\}. The position points (X, Y) are generated by random sampling and the velocity command v_{cmd} tells the hybrid controller to chose control strategy and use it as speed command.

### 6.2.3 Speed control

Since the sampling time, T_s, and the the speed v enters as a product in (6.4) one can see this product as a sampling distance, ds = T_s v. As mentioned earlier, the system characteristics are dependent on the size of the velocity. This means that the performance of the LQ controller will depend on size of the velocity. So instead of varying the sampling time we can vary the speed depending on how far the system is from its equilibrium configuration (4.18). To be far from equilibrium configuration is the same as having a big \( \alpha_c \) in (5.14). The open loop speed control we have used is

\[
v = v_{cmd} \left( 1.2 - \min \left( 1, \left( \frac{\alpha_c}{\alpha_{max}} \right)^2 \right) \right)
\]

where the speed command \( v_{cmd} \) is given from the planner and \( \alpha_{max} \) is the maximum steering angle. One can see this speed control as an extension in the closed-loop system proposed on the previous chapter. What (6.5) does is that it lowers the speed of the truck and trailer system when it is far from a desired equilibrium configuration. The speed will vary from 20 % to 120 % of the desired speed command, \( v_{cmd} \).

### 6.2.4 Sampling Strategies

Another extension proposed in [8] is that depending on the environment and the planners objective (Reverse parking, forward driving, etc.) we can speed up the CL-RRT algorithm by biasing the sampling. A randomly generated sample point (X, Y) in a geometric predefined region is performed by different sampling strategies and the two strategies used in this thesis is Uniform-sampling and Arc-sampling. Uniform-sampling with respect to some reference position (X_0, Y_0) is given by

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} = \begin{bmatrix}
X_0 + \sqrt{3} \sigma_X n_x \\
Y_0 + \sqrt{3} \sigma_Y n_y
\end{bmatrix}
\]

(6.6)

Where \( \sigma_X \) and \( \sigma_Y \) is the standard deviation in the respective coordinate axis. \( n_i \) is a uniformly distributed randomly generated number, i.e. \(-1 \leq n_i \leq 1\). Arc-sampling is performed in a similar way: Given a reference position and an orientation (X_0, Y_0, r_0, \theta_0) the sample point (X, Y) becomes

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} = \begin{bmatrix}
X_0 \\
Y_0
\end{bmatrix} + r \begin{bmatrix}
\cos \theta \\
\sin \theta
\end{bmatrix} \text{ where } \begin{cases}
\begin{align*}
\sigma_r &= \sqrt{3} \sigma_r |n_r| + r_0 \\
\sigma_\theta &= \sqrt{3} \sigma_\theta n_\theta + \theta_0
\end{align*}
\end{cases}
\]

(6.7)

where \( \sigma_r \) and \( \sigma_\theta \) is the standard deviation in radial and circumference direction. \( n_i \) is also here a uniformly distributed randomly generated number.
In each iteration the CL-RRT algorithm [11], [12] produces two samples, a random sample and a goal sample. Since the planning mission in this thesis is about reversing a truck and trailer around obstacles a sampling strategy has been developed that uses the fact that the truck and trailer can move both forwards and backwards. Combining these motions as a real truck driver would do; drive forward to arrive in a good configuration and then start to reverse. We have used Uniform-sampling for exploring and Arc-sampling for optimizing in a closer region of the goal. Since we had a bounded search area during the simulations, we wanted to use a quite general sampling strategy that solved as many different initial conditions as possible. The strategy for the random sampling is presented in Table 6.1 where \((x_{\text{mid}}, y_{\text{mid}})\) is the approximated midpoint of the map.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Direction</th>
<th>Bias</th>
<th>Standard deviation</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>Forward</td>
<td>((x_{\text{mid}}, y_{\text{mid}}))</td>
<td>(\sigma_x = 1.73) m, (\sigma_y = 1.73) m</td>
<td>0.3</td>
</tr>
<tr>
<td>Uniform</td>
<td>Backward</td>
<td>((x_{\text{mid}}, y_{\text{mid}}))</td>
<td>(\sigma_x = 1.15) m, (\sigma_y = 1.15) m</td>
<td>0.3</td>
</tr>
<tr>
<td>Uniform</td>
<td>Backward</td>
<td>((x_{\text{mid}}, y_{\text{mid}}))</td>
<td>(\sigma_x = 1.75) m, (\sigma_y = 1.75) m</td>
<td>0.4</td>
</tr>
</tbody>
</table>

*Table 6.1: Sampling strategy during random sampling.*

For generating a goal-sample near the goal point \(p_{\text{goal}} = [x, y, \theta]^T\), we used Arc-sampling with \((X_0, Y_0, r_0, \theta_0) = (x, y, -R, \theta)\) and \((\sigma_r, \sigma_\theta) = (R/9, \pi/17)\) where \(R\) is the look-ahead distance in backward motion.

### 6.2.5 Objective function

To generalize the optimal control problem (6.2) we need to specify what a trajectory costs. For this we have used a path length that just integrate the distance of the rear axle of the trailer [11], [12]. We have also made an additive final cost on the final configuration \(p(t_f) \in X_{\text{goal}}\). Given the ideal goal position \(p_g = [x_3, y_3, \theta_3, \beta_3, \beta_2]^T\) and a simulated goal position was \(p(t_f) = [x_3(t_f), y_3(t_f), \theta_3(t_f), \beta_3(t_f), \beta_2(t_f)]^T \in X_{\text{goal}}\) the objective function \(V\) becomes

\[
V = \sum_{i=1}^{N} \left( \sqrt{(x_3[i] - x_3[i-1])^2 + (y_3[i] - y_3[i-1])^2} + 10(\theta_3[N] - \theta_3,g)^2 + 10(\beta_3[N] - \beta_3,g)^2 + 10(\beta_2[N] - \beta_2,g)^2 \right)
\]

(6.8)

where \(N = t_f/T_s\). The additive final penalty is used to push the planner to choose trajectories that lead to a final configuration \(p(t_f)\) near the ideal configuration \(p_g\).

### 6.2.6 Obstacle Map

A big advantage with the RRT algorithm is the fact that it performs forward simulations of the system. For each time step during the forward simulation the
location of the vehicle is collision checked with an obstacle map. Information of all obstacles is defined in the Obstacle Map. The tree extension is marked feasible if the forward simulation was collision-free. Otherwise it is marked partly feasible or infeasible depending on how long the forward simulations was carried out before a collision was detected. An extension is then inserted in the tree if it turned out to be feasible or partly feasible.

### 6.2.7 Heuristics

The RRT algorithm expands the tree by performing a forward simulation from a suitable nearest neighbor to the new sampling point. The nearest neighbor is chosen based on some sort of heuristics. The evaluation of the heuristics is time consuming and therefore we are proposing that the heuristic map should be calculated offline. We have therefore simulated the system to different locations from an initial configuration where we have assumed all states to be zero in the local frame of the vehicle and then evaluated the extension based on different criteria. Firstly, if an extension did not succeed it is marked infeasible, otherwise depending on the direction of motion evaluated differently. As we want the RRT to generate smooth tracks we have decided to weight both distance and angle changes into the cost of an extension in backward motion $\Gamma_{\text{ref}}$, (6.9a) and to increase movability only weight the distance changes into the cost of an extension in forward motion $\Gamma_{\text{fwd}}$, (6.9b).

\[
\Gamma_{\text{ref}} = \sum_{i=1}^{N} \left( 10(x_3[i] - x_3[i-1])^2 + 10(y_3[i] - y_3[i-1])^2 + 10^4(\beta_3[i] - \beta_3[i-1])^2 + 10^4(\theta_3[i] - \theta_3[i-1])^2 \right) \\
\Gamma_{\text{fwd}} = \sum_{i=1}^{N} \left( 10(x_3[i] - x_3[i-1])^2 + 10(y_3[i] - y_3[i-1])^2 \right)
\]

How to generate the heuristic cost is described in Algorithm 1. For the Lego platform we did forward simulations over a grid where $-3.3 \, \text{m} \leq x_3 \leq 3.3 \, \text{m}$ and $-3.3 \, \text{m} \leq y_3 \leq 3.3 \, \text{m}$ with an accuracy of 3 cm. The resulting heuristic cost in forward motion can be seen in Figure 6.1a and in backward motion can be seen in 6.1b where the uniform dark blue marks infeasible extensions and then from dark blue to yellow marks rising extension cost.

### 6.2.8 The CL-RRT algorithm

Based on the previous discussion in this section we are now ready to describe the CL-RRT algorithm [8], [11], [12]. The algorithm is presented in Algorithm 2. The first thing that happens is that the planner is initiated with an objective, i.e take the vehicle to a goal point $p_g$ with constraints on the state configuration, $p(t_f) \in X_{\text{goal}}$ (line 1). After the objective has been initiated the planner will enter the planning-loop (line 2-35) and will remain there until the goal region has been reached. The planning-loop starts by updating information of the vehicle configuration and the Obstacle Map (line 3). Based on this new information the planning tree is updated by moving the root of the tree with the vehicle and hence old
Algorithm 1 Heuristic cost evaluation:

1: Grid up the area to evaluate
2: repeat
3: \hspace{1em} Initialize the vehicles initial states
4: \hspace{1em} Send a goal input to the closed-loop system
5: \hspace{1em} repeat
6: \hspace{2em} Perform a step on the closed-loop system
7: \hspace{2em} until Goal reached or time limit is reached
8: \hspace{1em} if Goal reached then
9: \hspace{2em} Evaluate extension cost
10: \hspace{1em} else
11: \hspace{2em} Mark extension as infeasible
12: \hspace{1em} end if
13: until Grid has been covered.
14: Save the result in a file.

(a) Simulated heuristics in forward motion. (b) Simulated heuristics in backward motion.

Figure 6.1: The heuristics of the closed-loop system in forward motion (a) and in backward motion (b).
Algorithm 2 CL-RRT Algorithm [8], [11], [12]:

1: Initiate the planning tree and define a goal point and a goal region.
2: repeat
3:   Update vehicle states and Obstacle Map
4:   Update the tree and delete old nodes
5: repeat
6:   Generate a random sample for the input to the controller
7:   Generate a goal sample for the input to the controller
8:   Generate a random variable between 0 and 1 called Optimize
9:   if Optimize > 0.5 then
10:      Select the nearest node in the tree using some heuristics
11:   else
12:      Select the cheapest node in the tree using some heuristics added with its previous trajectory cost
13: end if
14: Propagate from the selected node to the random sample
15: if Propagation succeeded then
16:   Update cost-to-go
17:   Propagate from random sample to goal sample
18: if Propagation succeeded then
19:   Update cost-to-go
20: if Goal region reached then
21:   Add final cost and mark the expansion as a possible solution
22: end if
23: end if
24: Insert all feasible and partly feasible nodes in the tree
25: until Time limit is reached
26: if Possible solution exists then
27:   Sort all possible solutions
28:   Choose the cheapest solution and do a recheck on its feasibility
29: if The best trajectory is infeasible then
30:   Remove the infeasible node from the tree and go to line 27.
31: end if
32: Send the best trajectory to the controller
33: end if
34: until Vehicle reaches the goal region.

that has the lowest trajectory cost added with the heuristic cost. If a propagation is succeeded then the cost-to-go is updated (line 16) and a propagation from the random sample to the goal sample is performed (line 17). This propagation is also collision checked and feasibility marked (line 18). If the propagation succeeded the cost-to-go is updated (line 19) and the goal sample propagation is checked if it reached the goal region. If the propagation reached the goal region it is added as a solution candidate and a final cost is added (line 20-21). All nodes that are
feasible or partly feasible is inserted in the planning tree (line 25). The tree expansion will continue until the time limit has been reached and then the planner will check if solutions exist (line 27). If solution candidates exist they are sorted based on their cost-to-go (line 28). The best candidate is chosen and its feasibility is rechecked (line 30) to make sure that the obstacle map has not changed since the trajectory was generated. If the solution is still feasible the corresponding reference control sequence is send to the controller of the truck and trailer system (line 33). If the best candidate turned out to be infeasible the node is deleted from the tree and if more solution candidates exist the procedure of sorting and rechecking is done again.

Using this algorithm a tree of reference signals is created. One such tree can be seen in Figure 6.2.

Figure 6.2: The implemented CL-RRT algorithm is solving a planning problem in a maze-like environment. The green circle is the goal region, the planning tree is the red and blue lines, the yellow line is the best solution with its corresponding trajectory, the light blue line. Obstacles are marked with black and red boxes.
In this chapter the main results from this thesis work will be presented. Section 1 contains an evaluation of the reference tracking performance for the closed-loop system in backward motion when using a smooth reference path. The evaluation is performed on the Lego platform described in Chapter 3. Section 2 contains some interesting planning results from the implemented CL-RRT algorithm in simulations. A sensitivity and robustness analysis of the closed-loop system when having geometric model errors will be covered in Section 3 and 4. Section 5 contains a successful experiment from the lab where the CL-RRT motion planning algorithm is used on the Lego platform described in Chapter 3.

7.1 Reference tracking of the closed-loop system

In this section we will analyze the reference tracking ability of the closed-loop system when driving backwards. We will also analyze the reference tracking ability of the LQ controller. For the closed-loop system the reference signal is the midpoint of the rear axle of the trailer \((x_3, y_3)\) and for the LQ controller the reference signal is relative angle \(\beta_3\). In the previous chapter we chose to have a quite large look ahead distance to be able to handle non-smooth reference signals without losing stability.

In this section we will analyze the result when choosing a more appropriate reference signal. We have constructed a reference signal that looks like an infinity shaped path (see Figure 7.1). The reference path consists of two half circles and two sinusoids. When having a smooth reference path we were able to lower the look ahead distance to \(R_{rev} = 0.66 \text{ m}\) and therefore increase the tracking performance. Using the extension proposed in (5.9) with \(K_p = 0.3\) we were able to achieve even better reference tracking performance. The result was logged and
the reference tracking can be viewed in Figure 7.1. The steering angle and the relative angles $\beta_2$ and $\beta_3$ together with their desired angles is plotted in Figure 7.2.

![Infinity path](image)

**Figure 7.1:** The Lego platform is driving in an infinity shaped path. The red line is the reference signal for the closed-loop system and the blue line is the output signal, e.g. the mid position of the rear axle of the trailer. In this figure we have displayed three laps to illustrate the accuracy.

From Figure 7.1 we can conclude that the reference tracking is quite good and in this experiment the Lego platform totally drove ten laps in a row with the same accuracy. When doing this experiment the result can vary a little based on the size of the bias in the steering mechanism. From Figure 7.2a we can conclude that modeling the backlash in the steering mechanism was well motivated since the same oscillation characteristics can been seen here as in Figure 5.8 (blue line). Note that in Figure 7.2a is a plot of the steering angle input (blue line) and not the actual steering angle. The actual steering angle is hard to measure online. In Figure 7.2 one can see that the LQ controller is able follow its reference signal, $\beta_3$, quite well. Due to the fact that the equilibrium configuration only holds in a stationary turn and the reference signal for the LQ controller is continuously changing, $\beta_2$ is often far for its equilibrium value.
7.1 Reference tracking of the closed-loop system

(a) The steering angle input is here plotted together with the desired steering angle. The solid red lines marks the saturation limits in the steering angle.

(b) The relative angle $\beta_2$ is here plotted together with its desired equilibrium value.

(c) The relative angle $\beta_3$ is here plotted together with its desired equilibrium value. $\beta_3$ is the reference signal for the LQ controller.

Figure 7.2: The Lego platform is driving in an infinity shaped path, see Figure 7.1. In this figure the relative angles (b),(c) and the steering angle (a) are plotted together with their desired values. Note that $\beta_3$ is the actual reference signal for the LQ controller. In this figure we have displayed one lap.
7.2 Evaluation of CL-RRT algorithm in simulations

This section contains some planning results for the CL-RRT algorithm. The planning problem (see Figure 7.3) consists of finding a path for the system through a maze-like environment to a goal region. This region introduced constraints on the system's final state configuration. Some solutions from the planning problem can be seen in Figure 7.3, 7.4a and 7.4b. Two different initial configurations were used. In Figure 7.3 and 7.4a the same planning problem (starting forward) has been run with two completely different solutions. In Figure 7.4b the initial state $\theta_3$ of the system has been flipped 180 degrees compared to the other figures (starting backward). Since the sampling strategy is randomized the computation time until finding a solution will vary. We have therefore performed twenty simulations starting in backward motion and the same number of simulations starting in forward motion. We measured time and number of nodes before finding a solution, the result is summarized in Table 7.1.

<table>
<thead>
<tr>
<th>Start configuration</th>
<th>Number of tests</th>
<th>Failure</th>
<th>Number of nodes</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting forward</td>
<td>20</td>
<td>3</td>
<td>19200 ± 9500</td>
<td>33 s ± 14 s</td>
</tr>
<tr>
<td>Starting backward</td>
<td>20</td>
<td>0</td>
<td>16300 ± 13100</td>
<td>19 s ± 25 s</td>
</tr>
</tbody>
</table>

*Table 7.1: Summarizing data with average values and standard deviation from planning with CL-RRT algorithm.*

When the planner planned more than 2 minutes without finding a solution the simulation was terminated and marked as a failure. What is interesting is that the consumed time until finding a solution varied quite much. Sometimes it found a solution within 2 seconds and the next time it took almost a minute.

*Figure 7.3: The implemented CL-RRT algorithm solving a nontrivial planning problem in a maze-like environment in simulation. The yellow line is the best solution in the reference space. The red, blue and yellow lines are the trajectories of the rear axle of the trailer for different inputs. Obstacles are marked with black and red lines.*
7.2 Evaluation of CL-RRT algorithm in simulations

(a) In this planning example the system is starting in forward motion. A completely different solution compared to Figure 7.3 with the same sampling strategy.

(b) In this planning example the system is starting in backward motion.

Figure 7.4: The implemented CL-RRT algorithm solving a nontrivial planning problem in a maze-like environment in simulation. The yellow line is the best solution in the reference space while the red, blue and yellow lines are the trajectories of the rear axle of the trailer for different inputs. Obstacles are marked with black and red lines.
### 7.3 Geometric sensitivity analysis

An important aspect when evaluating performance of a controller is to investigate how well it handles errors in the model. We will in this section investigate how the reference tracking performance of the closed-loop system will decrease when the trailer’s length in the model does not coincide with the trailer’s length in the system that is being controlled. In order to perform this investigation simulations will be done with wrong trailer length. The modeled trailer length is set to either 30% smaller or 30% larger than the real trailer’s length. We have also set the trailer length wrongly in the Pure pursuit controller to investigate a worst case scenario. The proportionality control of $\beta_3$ was set to zero ($K_p = 0$) in the Pure pursuit controller. To evaluate performance the same simulation test as in Chapter 5 will be done which consists of three 90 degree turns. Time constant ($T_{steer} = 1$ s), bias ($\alpha_{bias} = 0.05$ rad) and backlash ($b = 0.05$ rad) in the steering mechanism was modeled as before. In Figure 7.5 we have plotted the reference path together with the output $(x_3, y_3)$ for the closed-loop system when simulating the system with designed controllers for 70%, 100% and 130% of the correct trailer length. As you can see the reference tracking is deteriorated but the closed-loop system is still able to locally stabilize the system around the reference path. This result was expected because both the Pure pursuit controller and the gain scheduled LQ controller are constructed based on geometric properties of the vehicle. Note that, however, since the reference signal is not smooth also the controller with the correct trailer length is unable to follow the reference path particularly well at the kinks.

![Simulation of truck and trailer system.](image)

*Figure 7.5: Sensitivity analysis of the closed-loop system when having a geometric error in the length of the trailer. In this figure we have plotted the reference path together the output $(x_3, y_3)$ for the closed-loop system.*

In Figure 7.5 we have analyzed the sensitivity of the closed-loop system when having errors in the trailer’s length. From the same simulations it is also possible
to analyze the performance of the gain scheduled LQ controllers separately. Figure 7.6 contains plots where the reference signal for the LQ controller, $\beta_{3,d}$, has been plotted together with the output signal, $\beta_3$, during the same simulation as in Figure 7.5. Generally LQ control is known for having high tolerances against model errors [19], [20]. In Figure 7.6 the reference tracking performance of the LQ-controller can be viewed. From Figure 7.6a we can conclude that the output signal is magnified compared to the reference signal when the correct length is 70% of what is modeled. From Figure 7.6c we can see that the output signal is now instead damped and does not reach the desired height when the correct length is 130% of what is modeled. From Figure 7.6b we can conclude that even with correct trailer length the LQ controller can not perfectly follow the reference signal. This is partly because the reference path for the closed-loop system is non-smooth and partly due to the fact that gain scheduled LQ controller is controlling an unstable system with hard dynamics. From the plots we can conclude that the LQ controller handled this model error quite well, especially when the reference is changing slowly.

(Figure 7.6: Sensitivity analysis of the LQ-controller when wrong trailer length is modeled. Here $\beta_{3,d}$ is plotted together with $\beta_3$.)

(a) The correct length is 30% shorter than modeled.

(b) Correct length is modeled.

(c) The correct length is 30% longer than modeled.
7.4 Geometric robustness analysis

How well a controller handles model errors is often evaluated in terms of stability robustness. Previous section contained a sensitivity analysis of the closed-loop system when wrong trailer length was modeled. An equally important aspect is what happens with the stability of the closed-loop system with this model error. In Chapter 5 a numeric stability analysis of the closed-loop system was performed by simulating the system from different initial configurations in the relative angle $\beta_2$ and $\beta_3$. To analyze robustness the same types of simulations can be performed. This has been done identically to the stability analysis in Chapter 5 when the correct length is either 70 % or 130 % of what is modeled. The stability region can be analysed by varying the initial states of the relative angles and evaluate if the system is able to recover to follow a straight line. The region of

![Region of attraction](image1)

(a) The correct length is 30 % shorter than modeled.

![Region of attraction](image2)

(b) Correct length is modeled.

![Region of attraction](image3)

(c) The correct length is 30 % longer than modeled.

Figure 7.7: Simulated stability region of the closed-loop system following a straight line in backward motion. The same controller has been used on different trailer lengths.
attraction for different trailer lengths is plotted in Figure 7.7. From this figure one can conclude that the stability region is increasing with longer trailer. This is expected since a longer trailer implies a less aggressive system. This can be seen from model (4.13) where a higher value on $L_3$ implies that both $\hat{\beta}_3$ and $\hat{\theta}_3$ will be smaller. In aspect of stability robustness it is better to use a longer trailer than the modeled compared to use a shorter trailer than the modeled. It is important to understand that the region of attraction is both a matter of control design and a system property. The best is always to use a controller design for the correct geometric combination.

7.5 Autonomous parking of the Lego platform

In this last section we will show a successful result where the CL-RRT algorithm is planning a route for the Lego platform described in Chapter 3. When a solution is found the reference path is transmitted to the LEGO-NXT via Bluetooth. The Pure pursuit controller and the LQ controller were implemented on the LEGO-NXT. The design parameters for the Pure pursuit controller was set to $R_{fwd} = 60$ cm, $R_{rev} = 100$ cm and $K_p = 0.1$. A photo of the lab can be seen in Figure 7.8. The area is rectangular (2 m x 3 m), bounded by walls and the area has two small quadratic obstacles. Figure 7.8 is from the implemented CL-RRT algorithm where a solution has been found (the yellow line). One such solution has been transmitted to the LEGO-NXT and the execution of the reference path is shown in Figure 7.9. In Figure 7.10 we have plotted the steering angle $\alpha$ and the relative angles $\beta_2$ and $\beta_3$ together with their desired equilibrium values given by the Pure pursuit controller. One thing to mark is that some other experiments

![Figure 7.8](image.jpg)

**Figure 7.8:** The implemented CL-RRT algorithm is planning a parking maneuver for the Lego platform. The planning tree is the red and blue lines and the yellow line is the best solution. Obstacles are marked with black and red boxes.
**Figure 7.9:** The Lego platform is following a reference path planned by the CL-RRT algorithm. The reference path is transmitted via Bluetooth.

did not end up in such a good final configuration as in this example because the Lego platform did not follow the planned reference path as intended. There are at least two reasons for this. The first reason is because we did not model the backlash nor the bias in the steering mechanism for the planners simulation model (6.4), i.e. we had model errors. The second reason is that we didn’t update the reference path as the LEGO-NXT executed the reference path. If we had updated the reference path based on feedback from reality, the planner would be able to compensate for eventual deviation from the originally planned trajectory.
(a) The desired steering angle command from the pure pursuit controller is plotted together with the steering angle input. The solid red lines marks the saturation limits in the steering angle.

(b) The relative angle $\beta_2$ is plotted together with its desired equilibrium value.

(c) The relative angle $\beta_3$ is plotted together with its desired equilibrium value.

**Figure 7.10:** The Lego platform is following a reference path planned by the CL-RRT algorithm. In this figure the relative angles (b), (c) and the steering angle (a) are plotted together with their desired values. Note that $\beta_3$ is the actual reference signal for the LQ controller.
In this thesis a control strategy to stabilize and control a truck and trailer system is presented. Also a motion planning algorithm for the system has been implemented. The use of CL-RRT motion planning algorithm for this particular system is the main contribution of this thesis work. In this chapter we will summarize the contributed results and discuss future work.

8.1 Conclusions

This thesis work has been treating modeling, controlling and motion planning of a truck and trailer system in backward and forward motion. For modeling we have used and derived a general recursive formula to describe a general N-trailer system. Therefore the work done in this thesis can be used as a platform for other truck and trailer systems. Since the model holds under the assumption of rolling without slipping of the wheels the model is correct during low-speed maneuvers and when the surface has acceptable friction. A major problem and a major insight is the impact of a possible backlash and/or bias in the steering mechanism. The conclusion is that if backlash and/or bias exists, they should not be neglected because they introduce nonlinearities in the model that affects the design of the controllers.

The control strategy proposed in this thesis is based on geometric properties for the truck and trailer system. This means that it is straight forward to use the proposed controllers on any other truck and trailer system with other geometric properties. The only thing that may be needed is to adjust the controller parameters. We have shown that the LQ controller and the Pure pursuit controller can handle quite big geometric errors in the trailers length (±30 %) without loosing stability. However, the reference tracking performance was slightly reduced.
Therefore planning with the CL-RRT algorithm will not work with big geometric errors such as wrong trailer length. The planner should be able to handle small geometric errors but since the forward simulated trajectory of the closed-loop system is collision checked, the trajectory has to be accurate to ensure that a planned route is collision free.

If the reference path for the closed-loop system was chosen as a smooth function the reference tracking performance was very good on the Lego platform (see Figure 7.1). We have also shown that adding a proportionality control on the Pure pursuit controller in backward motion increased the tracking performance.

The CL-RRT algorithm was implemented and used for motion planning. It has shown promising performance in simulations where it has planned tricky maneuvers even through a maze-like environment by combining backward and forward motion. Since the motion planning algorithm is randomized the computation time varied. When we have used the CL-RRT algorithm on the lego platform it has sometimes worked but it has not shown robustness. We think that it is partly an implementation issue and partly because of the nonlinearities in the steering mechanism that we have not modeled in the simulation model used by the CL-RRT algorithm.

8.2 Future work

Since the control strategy has shown to work well on the Lego platform in the normal case a natural next step would be to implement the controllers on a full scale truck and trailer. When implementing the system on a full scale truck and trailer a new measuring system has to be implemented. Also different types of control strategies would be interesting to evaluate and compare. It would be interesting to add an integral path in the LQ controller which hopefully can decrease the impact of the bias in the steering mechanism and also replace the proportionality controller.

The Lego platform had some issues with backlash and bias in the steering mechanism. On a full-scale truck and trailer system these properties can be more or less neglected. This will have positive impact on the performance of the closed-loop system. Furthermore, the robustness and sensitivity properties of the low-level controllers have to be further investigated. It would also be interesting to on-line estimate the actual length of the trailer.

Regarding future work on the implemented CL-RRT algorithm, the heuristic cost-map in backward motion should be split up in more cases depending on the relative angles $\beta_2$ and $\beta_3$. For example, now we do not take into account that it may be more costly to connect from a node with final states $\beta_2 = 0.8$ rad and $\beta_3 = -0.9$ rad than if the states were $\beta_2 = \beta_3 = 0$ in backward motion. One other important aspect when choosing the nearest neighbor based on heuristics is the
fact that right now we do not take into account that the obstacle map is known. As it is now a nearest neighbor can be on the other side of a wall and their is no possibility for a connection. This can be one reason why the CL-RRT algorithm sometimes didn't find a solution in the maze example (see Table 7.1).

When the CL-RRT algorithm was used as a motion planner for the lego platform it planned in open-loop. In future work, replanning on-line should be pertained.
Bibliography


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Oskar Ljungqvist