Bachelor's Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Pallet Truck Trajectory Design and Control

Adam Basl

Supervisor: Vladimír Smutný, Ph.D.
May 2024

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Basl Adam**	Personal ID number:	**477363**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Pallet Truck Trajectory Design and Control

Bachelor’s thesis title in Czech:

Řízení paletového vozíku do cílové polohy

Guidelines:

1. Get familiar with the task of automatic loading of pallet using a pallet truck and information from the camera system.
2. Survey trajectory design methods for pallet truck and its control along the designed trajectory. Focus on the visual servoing approach where the trajectory is modified in the feedback loop depending on the camera data.
3. Implement selected algorithms in Robot Operating System 2. Focus on the control part of the algorithms.
4. Test the developed system in simulation and during navigation of the real pallet truck if it’s hardware will be available.
5. Evaluate the results.

Bibliography / sources:

1.] A. Mohammadi, I. Mareels and D. Oetomo, "Model predictive motion control of autonomous forklift vehicles with dynamics balance constraint," 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 2016, pp. 1-6, doi: 10.1109/ICARCV.2016.7838833.
2.] Tamba, T.A., Hong, B. & Hong, KS. A path following control of an unmanned autonomous forklift. Int. J. Control Autom. Syst. 7, 113–122 (2009). https://doi.org/10.1007/s12555-009-0114-y
3.] Vicent Girbés and Leopoldo Armesto and Josep Tornero Path following hybrid control for vehicle stability applied to industrial forklifts, Robotics and Autonomous Systems, Volume 62, Issue 6, 2014, Pages 910-922 https://doi.org/10.1016/j.robot.2014.01.004
4.]Shuping Chen, Huiyan Chen, Dan Negrut. Implementation of MPC-Based Trajectory Tracking Considering Different Fidelity Vehicle Models[J]. JOURNAL OF BEIJING INSTITUTE OF TECHNOLOGY, 2020, 29(3): 303-316. DOI: 10.15918/j.jbit1004-0579.19101

Name and workplace of bachelor’s thesis supervisor:

Ing. Vladimír Smutný, Ph.D. Robotic Perception CIIRC

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **02.02.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

Ing. Vladimír Smutný, Ph.D. Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

I want to express my heartfelt gratitude to everyone who supported me throughout the completion of this bachelor's thesis. My deepest appreciation goes to my advisor, Vladimír Smutný, for his invaluable guidance, feedback, and support during the research and development process and great thanks to Miroslav Uller who helped me a lot with the software part of the thesis and was always willing to help.

I am also very thankful to my family and friends for their enduring support and motivation. Their love and encouragement kept me driven to complete this thesis and reach this significant milestone.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 24. May 2024

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 24. May 2024

Abstract

In this thesis, our primary objective was to design the trajectory for an autonomous pallet loading truck and subsequently develop its control system.

The first part of the thesis focuses on analyzing the problem of autonomous pallet picking. Next, we delve into the properties of clothoid curves as potential trajectories for the pallet truck. These curves are thoroughly explored in terms of their suitability for out task of autonomous pallet picking.

In the second part of the thesis, we introduce the Pure Pursuit Controller (PPC), a method used to guide the truck along the planned trajectory. The PPC's sufficient usability is demonstrated in the simulation through several scenarios, each with different properties and challenges.

All the work was carried out with the real-world application in mind, ensuring that the transition from simulation to actual implementation will be as easy as possible.

Keywords: Clothoid, Visual servoing, autonomous pallet picking, Pure Pursuit Controller, ROS2

Supervisor: Vladimír Smutný, Ph.D.
CIIRC,
Jugoslávských partyzánů 1580/3 ,
Praha 6

Abstrakt

V této práci bylo naším hlavním cílem navrhnout trajektorii pro autonomní paletový vozík a následně vyvinout jeho řídicí systém.

První část práce je zaměřena na analýzu problému autonomního nakládání palet. Dále se zabýváme vlastnostmi clothoid jako potenciálních trajektorií pro paletový vozík. Tyto křivky jsou důkladně prozkoumány z hlediska jejich vhodnosti pro úlohu autonomního vychystávání palet.

Ve druhé části práce představujeme Pure Pursuit Controller (PPC), metodu používanou k vedení vozíku po plánované trajektorii. Dostatečná použitelnost PPC je demonstrována v simulaci prostřednictvím několika scénářů, z nichž každý má jiné vlastnosti a problémy.

Veškerá práce byla prováděna s ohledem na reálné použití, což zajišťuje, že přechod od simulace ke skutečné realizaci bude co nejjednodušší.

Klíčová slova: Clothoid, Visual servoing, autonomní nabírání palet, Pure Pursuit Controller, ROS2

Překlad názvu: Řízení paletového vozíku do cílové polohy

Contents

1 Introduction **1**

2 State of the art **3**

2.1 Tracking of the trajectory 3

2.1.1 Other related works 4

2.2 Visual servoing 5

3 Technical equipment **7**

3.1 Pallet truck 7

3.1.1 Dimensions 7

3.1.2 Kinematics 9

3.1.3 Motion of the pallet truck . . . 10

3.1.4 Motorization 11

3.2 Pallet 12

3.2.1 Dimensions 12

4 Problem analysis **15**

4.1 Localization 15

4.2 Trajectory planning 16

4.2.1 G^1 fitting with clothoids 17

4.2.2 G^2 fitting with clothoids 19

4.2.3 Visual servoing restrictions . . 22

5 Control **25**

5.1 Different sampling speeds of the control loops 25

5.2 Dealing with the delay 27

5.3 ROS scheme 28

5.4 Pure Pursuit Controller 28

6 Simulation **33**

6.1 Setup 33

6.2 Simulation operation 34

6.2.1 Odometry calculation 35

6.3 Simulation results 38

6.3.1 Simplest case - following single G2 clothoid 38

6.3.2 Planned path updates in low frequency 39

6.3.3 Considering noisy pallet pose estimate 40

6.3.4 Planned path updates in high frequency 41

6.3.5 Rate of change in angular velocity is too high 41

7 Conclusion **43**

A Software libraries **45**

B Used AI and other supportive tools. **47**

C Bibliography **49**

Figures

2.2 Smooth planned trajectory. 4

3.1 Dimensions of the pallet truck. . . 7

3.2 Our pallet truck. 8

3.3 Pallet truck's frames. 9

3.4 Kinematics of the pallet truck. The bicycle model is used for the derivation of the kinematics. 10

3.5 Rotation around ICR. 11

3.7 Pallet description. 12

3.8 Desired pose in front of the pallet. 13

4.1 Errors in the pallet pose estimation. 16

4.2 Planned trajectory to desired pose in front of the pallet. 17

4.3 G^1 clothoid with starting pose $(0,\ 0,\ 0°)$ and ending pose $(5,\ 4,\ 10°)$ with constant rate of change of curvature. 18

4.4 G^1 clothoid trajectories for $\mathcal{O}_\mathcal{F}$ and $\mathcal{O}_\mathcal{W}$. 18

4.5 G^1 clothoid steering angles and curvatures. 19

4.6 G^2 clothoid. 20

4.7 G^2 clothoid with a non zero starting curvature κ_s 21

4.8 FOV restrictions. 22

4.9 Invalid trajectories. 22

4.10 Permissible starting positions. . 23

4.11 Permissible starting poses. 23

5.1 Control Scheme. 26

5.2 Composition of the delay. 27

5.3 Delayed pallet pose estimate. . . . 28

5.4 ROS communication scheme. . . . 29

5.5 Geometry of PPC 30

6.1 Simulation scene setup. 33

6.2 Simulation visualization. 34

6.3 Simulated camera and depth sensor. 34

6.4 ROS communication in simulation. 35

6.5 Rviz comparison of odometries. . 37

6.6 Single G^2 clothoid. 38

6.7 Simplest case steering angles comparison. 38

6.8 Multiple trajectories. 39

6.9 Low frequency trajectory updates. 39

6.10 Trajectories with noisy pallet estimates. 40

6.11 Low frequency trajectory updates considering noisy pallet pose estimates. 40

6.12 High frequency updated trajectories with noisy pallet estimates. 41

6.13 High frequency trajectory updates considering noisy pallet pose estimates. 41

6.14 High frequency updated trajectories with noisy pallet estimates. 42

6.15 Control inputs with filtered γ_d. 42

Chapter 1

Introduction

This bachelor's thesis is part of a larger project aimed at developing a system for automatic pallet detection and subsequent automatic pallet loading and unloading. The system will be used in conjunction with a construction robot from KM Robotics [21] to enhance automation and reduce the need for human labor.

The Idea is that the human operator will take the pallet truck close to the pallets loaded with bricks. The pallet truck, equipped with sensors, will detect individual pallets and allow the operator to select a pallet through a graphical user interface (GUI). The pallet truck will then autonomously load the selected pallet, which the operator will transport to the vicinity of the construction robot.

The construction robot needs the pallet with bricks to be placed in exact position for him to be able to pick individual bricks with which he is building the wall. The Pallet truck and construction robot will communicate with each other and the robot will tell the pallet truck where to place the pallet with the bricks.

These tasks involve similar processes. During the loading phase, the sensors on the pallet truck will estimate the pallet's position. In the unloading phase, the position will be provided by the construction robot. The dynamics of the pallet truck will vary when carrying a loaded pallet.

In this thesis we are going to focus on the trajectory planning and control of the pallet truck. Visual systems and the processing of image and depth data is another major topic that Richard Randák addresses in his bachelor's thesis [20].

Chapter 2

State of the art

2.1 Tracking of the trajectory

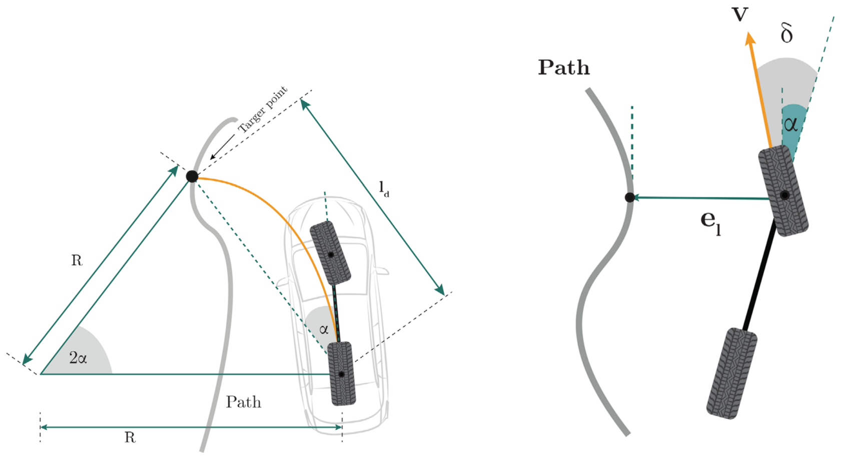
There are several ways ho to follow planned trajectory or so called path tracking controllers. There are several categories of the path tracking controllers [22].

Geometry based tracking controllers

First category is geometry-based controllers. In the first category the belong Pure Pursuit [24], [22] and Stanley controller. Those are quite simple but perform well in lower speeds which is our case.

MPC

Model predictive control (MPC) utilizes an explicit model of the system to estimate future states over a defined time horizon. At each sampling interval, the controller aims to find an optimized control sequence based on these predictions. Only the first control action from this sequence is implemented, and the process repeats at regular intervals. This methodology is detailed in [22], [13], and [15], which additionally discusses constraints related to vehicle dynamic balance—an important consideration for pallet trucks transporting heavy loads such as bricks.



(a) : PPC geometry. **(b) :** Stanley controller geometry.

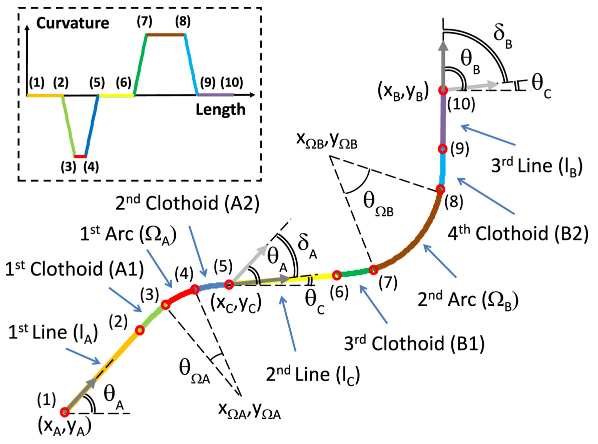


Figure 2.2: Planned trajectory proposed by [10]. Trajectory consists of several segments and each segment is either straight line, arc or clothoid.

2.1.1 Other related works

The similar problem of path tracking with the forklift is discussed at [23]. Other work [15] implements the MPC approach for the path tracking and ads constraints for the vehicles dynamics balance. The work [10] outlines a closed-loop hybrid controller, integrating kinematic and dynamic elements, designed for precise path following with industrial forklifts carrying heavy loads at high speeds. It prioritizes factors like vehicle stability, safety, wheel slippage.

2.2 Visual servoing

The principle of visual servoing involves utilizing visual feedback from a camera system to control the motion of a robot or a mechanical system. The basic idea is to adjust the system's movements based on visual information obtained from the visual sensors.

Visual servoing typically operates within a feedback loop, where the visual sensor continuously captures images of the environment, and these images are processed to extract relevant features or information. This information is then used to compute the desired motion commands for the robot or system in real-time.

There are two main approaches to visual servoing. One of them beeing Image-based visual servoing (IBVS) and the other Position-based visual servoing (PBVS). The IBVS usually computes control signals based directly on the visual features obtained by the visual sensors. On the other hand in the PBVS approach the control signals are computed based on the 3D features of the scene derived from the visual sensors. Both of these approaches for path following are discussed in [[4], [3].

Chapter 3

Technical equipment

3.1 Pallet truck

3.1.1 Dimensions

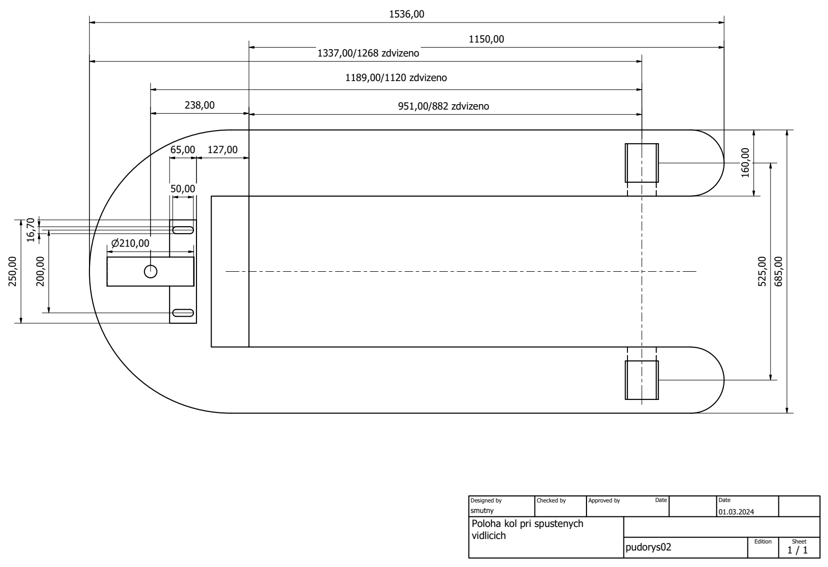
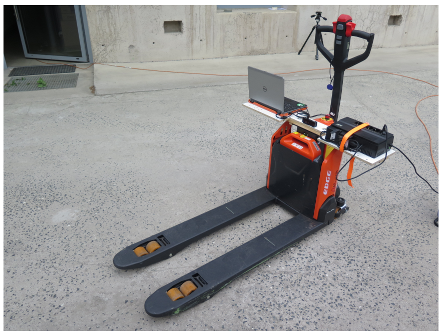


Figure 3.1: Dimensions of the pallet truck.

We are working with the pallet truck NOBLELIFT PTE20N. It's dimensions can be seen in figure 3.1

(a) : Pallet truck taken from the official site [18].



(b) : Pallet truck NOBLELIFT PTE20N mounted with RGBD camera, notebook and battery.

Figure 3.2: Our pallet truck.

Important parameter is the wheelbase distance $l_{wb} = 1189$ mm between axis of the front wheels and the rear wheel. Width of the pallet truck is $w_t = 685$ mm. Firstly we need to define important coordinate frames. Frame $\mathcal{F}$ is connected to intersection of the axis of the vehicle and the axis passing through front wheels in the fork of the pallet truck. Frame denoted as $\mathcal{W}$ is connected to the single motorized wheel at the rear of the pallet jack and the frame $\mathcal{C}$ is connected do the RGBD camera. The coordinate frames conncected to the pallet truck can be seen in figure 3.3.

The $T_{\mathcal{CF}}$ transformation between frames $\mathcal{C}$ and $\mathcal{F}$ is a constant one. Transformation $T_{\mathcal{WC}}$ depends on the steering angle γ of the rear wheel:

$$T_{\mathcal{CF}} = \begin{pmatrix} \boldsymbol{I} & \begin{pmatrix} l_{cam} \\ 0 \end{pmatrix} \\ \boldsymbol{0} & 1 \end{pmatrix}, \tag{3.1}$$

$$T_{\mathcal{CW}} = \begin{pmatrix} \boldsymbol{R}(-\gamma) & \begin{pmatrix} l_{wb} \\ 0 \end{pmatrix} \\ \boldsymbol{0} & 1 \end{pmatrix}, \tag{3.2}$$

where $\boldsymbol{R}(\gamma)$ denotes rotation matrix $\begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}$.

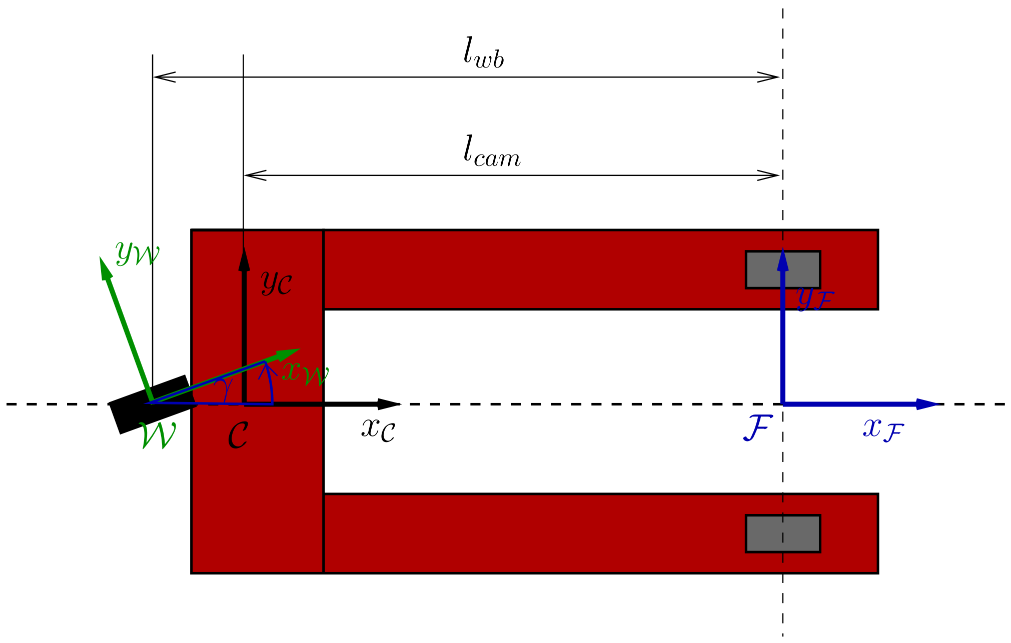


Figure 3.3: Pallet truck's frames. $\mathcal{F}$ is based in the intersection of the axis of the vehicle and the axis passing through front wheels in the fork. $\mathcal{C}$ Represents the pose of the RGBD camera and $\mathcal{W}$ is based in the center of the rear motorized wheel. Constants l_{wb} and l_{cam} represent the wheel-base distance and the distance between $\mathcal{F}$ and $\mathcal{C}$ respectively.

3.1.2 Kinematics

Diagram of our vehicle is shown in figure 3.4. There are two velocities we should consider. Velocity v_f of the frame $\mathcal{F}$ and velocity v_w of the frame $\mathcal{W}$ are:

$$v_{\mathcal{W}} = r_d \omega_d \,, \tag{3.3}$$

$$v_{\mathcal{F}} = v_{\mathcal{W}} \cdot \cos\gamma \,, \tag{3.4}$$

where r_d is the radius of the back motorized wheel and ω_d is it's angular velocity.

Then we can derive the motion equations with respect to reference frame $\mathcal{O}$:

$$\dot{x} = v_{\mathcal{F}} \cdot \cos\theta \,, \tag{3.5}$$

$$\dot{y} = v_{\mathcal{F}} \cdot \sin\theta \,, \tag{3.6}$$

$$\dot{\theta} = \frac{r_d \omega_d \cdot \sin\gamma}{l_{wb}} = \frac{v_{\mathcal{F}}}{l_{wb}} \cdot \tan\gamma \,. \tag{3.7}$$

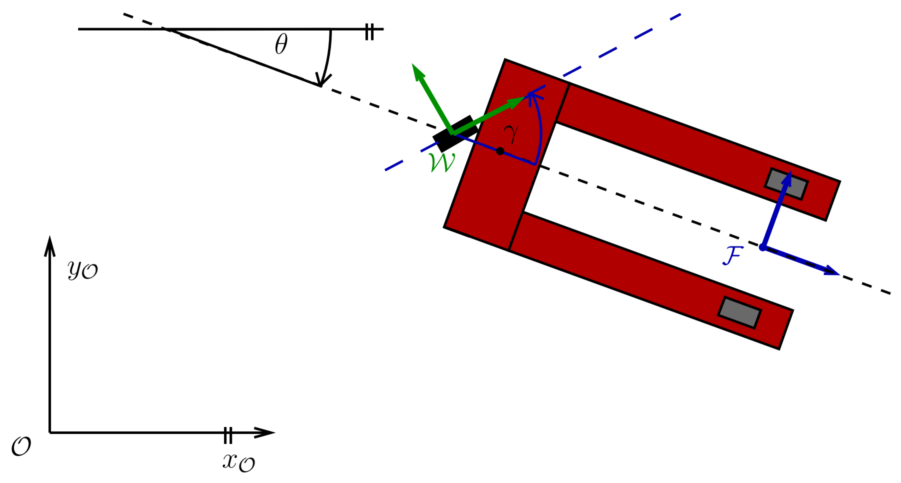


Figure 3.4: Kinematics of the pallet truck. The bicycle model is used for the derivation of the kinematics.

3.1.3 Motion of the pallet truck

When the steering wheel is turned at the steering angle γ the whole vehicle follows a circular path around a point called instantaneous center of rotation (ICR). This point is at the intersection of the front and rear wheel axes. We can calculate the radii ($r_{\mathcal{W}}$ and $r_{\mathcal{F}}$) of the circles with the center in the ICR passing through the frames $\mathcal{W}$ and $\mathcal{F}$ origins. The steering angle defines radii and also the curvatures ($\kappa_{\mathcal{W}}$ and $\kappa_{\mathcal{F}}$) around ICR as follows:

$$r_{\mathcal{W}} = \frac{l_{wb}}{\tan\gamma}\,, \tag{3.8}$$

$$r_{\mathcal{F}} = \frac{l_{wb}}{\sin\gamma}\,, \tag{3.9}$$

$$\kappa_{\mathcal{W}} = \frac{1}{r_{\mathcal{W}}}\,, \tag{3.10}$$

$$\kappa_{\mathcal{F}} = \frac{1}{r_{\mathcal{F}}}\,. \tag{3.11}$$

This means that in the situation when the steering angle γ approaches zero radii approach infinity and the curvatures approach zeros.

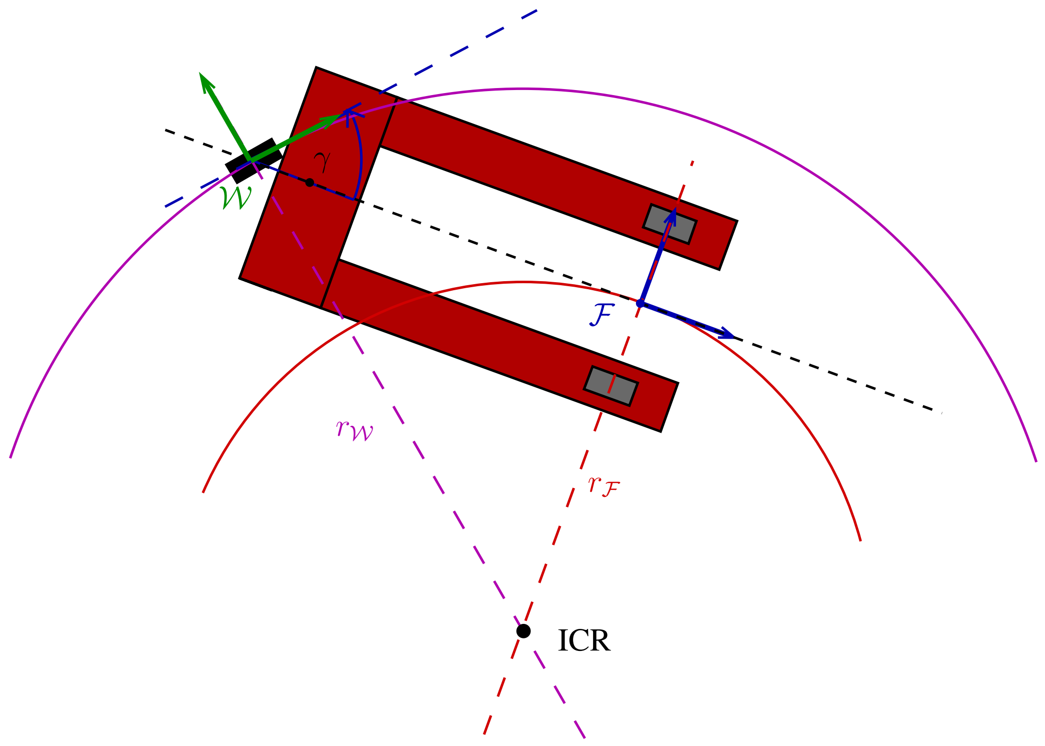


Figure 3.5: Given a steering angle γ the pallet truck rotates around ICR. Origins of frames $\mathcal{O}_{\mathcal{F}}$ and $\mathcal{O}_{\mathcal{W}}$ follow circular paths around ICR with radii $r_{\mathcal{F}}$ and $r_{\mathcal{W}}$.

3.1.4 Motorization

On the original unmodified pallet truck, only the travel on the rear wheel is motorized. The speed is controlled by the slider on the handle. The motor that turns the rear wheel is a 48V BLDC motor controlled by the Curtis 1226BL Motor controller. The communication between the drawbar and the Curtis motor controller is via CAN bus and the message with requested speed is send regularly approximately each 50 ms.

For the autonomous driving of the pallet truck, we also need to control the rotation of the rear wheel in other words the steering angle γ. For this purpose, a new motor and a new controller need to be integrated into the pallet truck. The motor is DMKE DB86-48 with the 1:140 gearbox, controlled by the ODrive S1 controller.

(a) : ODrive s1 take from [19].

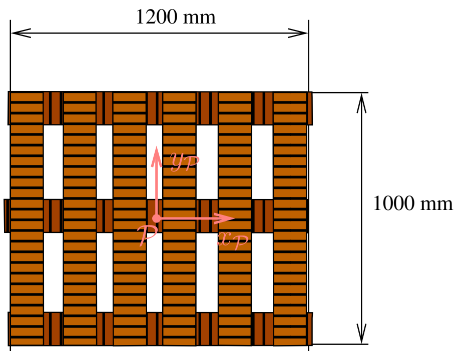
(b) : DMKE motor with gearbox.

3.2 Pallet

3.2.1 Dimensions

We need to establish frames connected to the pallet, which we want to load. The coordinate frame $\mathcal{P}$ is located at the center of the pallet. It's dimensions are pallet width $w_p = 1000$ mm and pallet length $l_p = 1200$ mm.

(a) : Pallet loaded with bricks.



(b) : Dimensions of the pallet.

Figure 3.7: Pallet description.

When we want to pick the pallet, our goal is to navigate the pallet truck to the desired pose connected to the frame $\mathcal{D}$ which is located in front of the pallet as displayed in figure 3.8.

The transformation $T_{\mathcal{D}\mathcal{P}}$:

$$T_{\mathcal{DP}} = \begin{pmatrix} \boldsymbol{I} & \begin{pmatrix} \frac{l_p}{2} + \frac{w_t}{2} \\ 0 \end{pmatrix} \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{3.12}$$

is constant so when we get the position of the pallet, we can easily calculate the desired pose for our pallet truck.

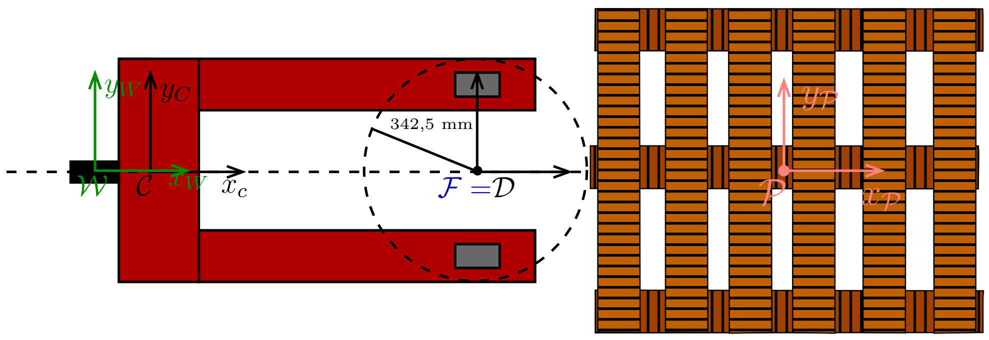


Figure 3.8: Desired pose in front of the pallet. In the case where the pallet truck is in the desired pose, the origins of the coordinate frames $\mathcal{O}_{\mathcal{F}}$ and $\mathcal{O}_{\mathcal{D}}$, along with their orientations, merge together.

Chapter 4

Problem analysis

4.1 Localization

First and important step the localization of the pallet. This task is difficult and is being handled by Richard Randák in his bachelor's thesis [20]. As an output of our vehicle's vision system we should get a relative estimated position of the pallet center to the $\mathcal{C}$. The estimates are getting better as we get closer to the pallet. At greater distances, there are going to be more significant errors in the pallet pose estimation. This situation is shown at the figure 4.1. Actual pose of the pallet at the camera frame $\mathcal{C}$ is defined by the distance of the pallet center from the camera $l_{\mathcal{C}}$, the azimuth angle δ and it's orientation ϕ. Estimated position might differ, represented by the estimated distance $l'_{\mathcal{C}}$ estimated azimuth angle δ' and estimated orientation ϕ'.

Experiments with the vision system, conducted by Richard Randák and discussed in [20], have shown that the error of the estimated pose of the pallet are relatively small regarding the estimated distance $l_{\mathcal{C}}$ and estimated azimuth δ. Errors in the estimated orientation ϕ' of the pallet are much more significant.

Due to the inaccurate and unreliable pallet pose estimates at larger distances, we cannot plan the path to the pallet in advance and follow it, as we would likely end up in the wrong location. Precision in the pallet loading is crucial. This leads us to the technique of visual servoing. It's principle involves continuously processing visual data and continuously updating the pallet pose estimate. The premise is that as the distance decreases, the pallet pose estimates get better and better and we can plan the trajectory precisely.

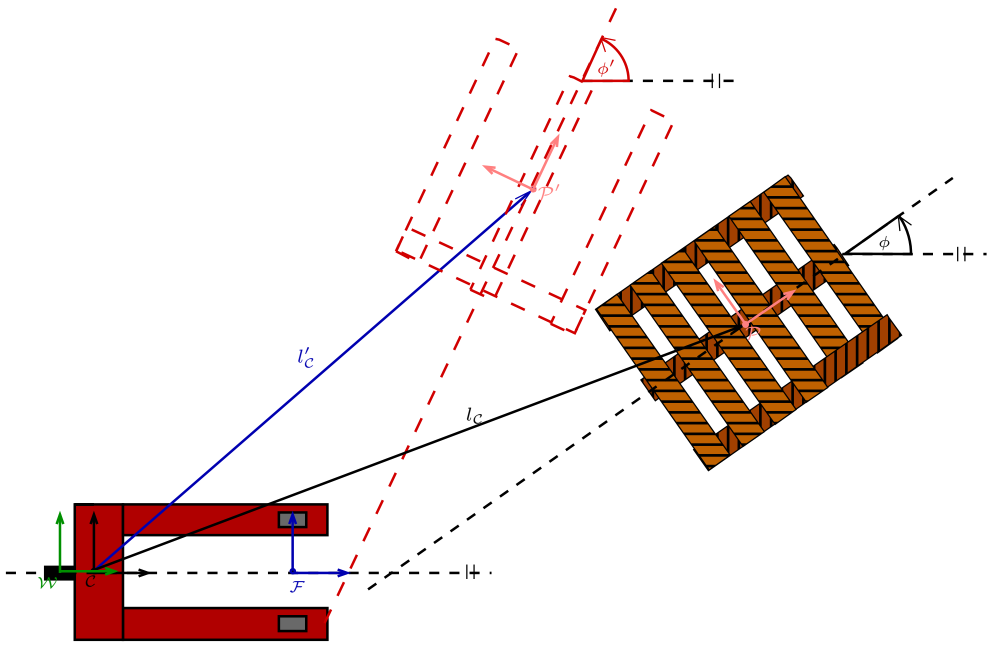


Figure 4.1: Errors in the pallet pose estimation. Actual pose of the pallet at the camera frame $\mathcal{C}$ is defined by the distance of the pallet center from the camera $l_{\mathcal{C}}$, the azimuth angle δ and it's orientation ϕ. Estimated position might differ, represented by the estimated distance $l'_{\mathcal{C}}$ estimated azimuth angle δ' and estimated orientation ϕ'. The most significant error occurs in estimated pallet orientation ϕ'. The reasons are discussed in Randák's bachelor's thesis [20].

4.2 Trajectory planning

Once we have the estimated pose of the pallet that we want to load, we can plan the trajectory along which the pallet truck should move. We are using clothoids. The main property of clothoid is that it's rate of change in curvature with respect to arc length is constant. This means that the curvature increases or decreases steadily as you move along the curve. How to calculate such curves is covered at the articles [5] and [6]. For calculations we are using the pyclothoid library [8].

Curvature κ and tangent angle τ at given arc-length of the clothoid can be calculated as follows:

$$\kappa(s) = \dot{\kappa}s + \kappa_0 \,, \tag{4.1}$$

$$\tau(s) = \frac{\dot{\kappa}}{2}s^2 + \kappa_0 s + \tau_0 \,. \tag{4.2}$$

Each clothoid with starting positions $(x_0,\ y_0)$ initial tangent angle τ_0 and initial curvature κ_0 can be parameterized by it's arc-length. s in a following form:

$$x(s) = x_0 + \int_0^s \cos\left(\frac{\dot{\kappa}}{2}s^2 + \kappa_0 s + \tau_0\right) ds\,, \tag{4.3}$$

$$y(s) = x_0 + \int_0^s \sin\left(\frac{\dot{\kappa}}{2}s^2 + \kappa_0 s + \tau_0\right) ds\,. \tag{4.4}$$

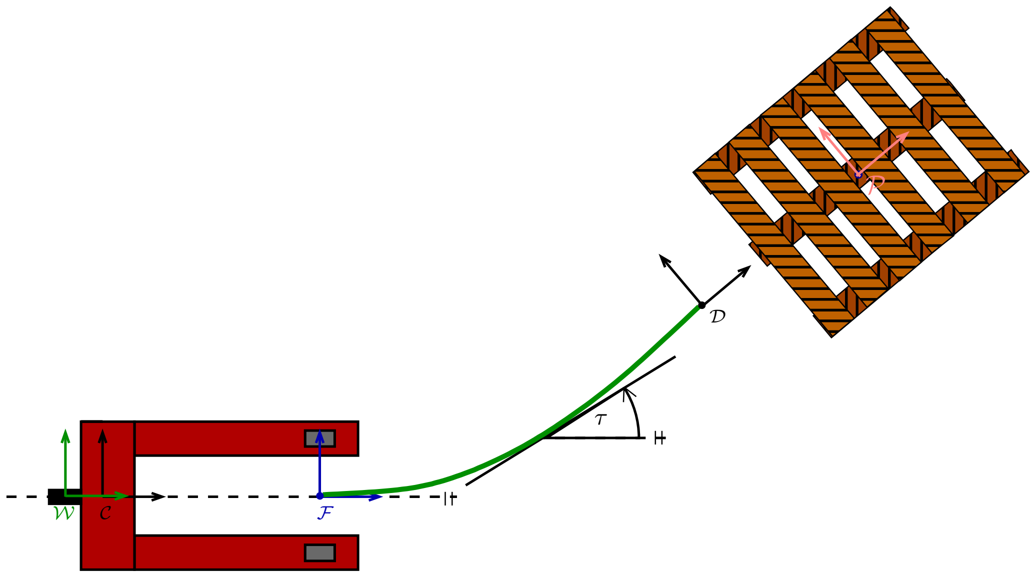
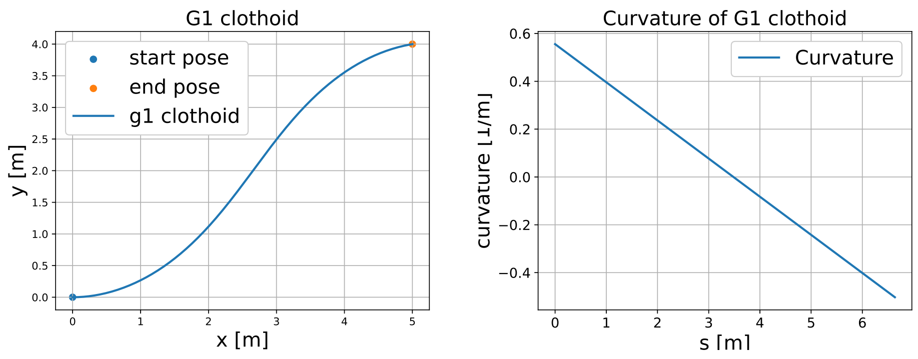


Figure 4.2: Planned trajectory to desired pose in front of the pallet. $\mathcal{O}_\mathcal{F}$ is at coordinates $(x_0,\ y_0)$ and its orientation is τ_0. $\mathcal{O}_\mathcal{D}$ is at coordinates $(x_f,\ y_f)$ and its orientation is τ_f. At any arc-length s a tangent angle τ can be calculated.

4.2.1 G^1 fitting with clothoids

Given a starting pose (x_0, y_0, τ_0) and an ending pose (x_f, y_f, τ_f), we can calculate a clothoid curve connecting these two poses. One disadvantage of these curves is that we can control only the tangent angles τ at the starting and ending points, but not the curvature values. This limitation means we cannot account for the current orientation of the steering wheel. Furthermore, the curvature of a G^1 clothoid at the end is not zero. As a result, while we would reach our desired pose $\mathcal{D}$, the steering angle γ also would not be zero. Consequently, before the final movement of the pallet truck underneath the pallet, we would need to adjust the steering angle once again.



(a) : G^1 trajectory. **(b) :** G^1 curvature.

Figure 4.3: G^1 clothoid with starting pose $(0,\ 0,\ 0°)$ and ending pose $(5,\ 4,\ 10°)$ with constant rate of change of curvature.

In figure 4.3 a simple G^1 clothoid connecting starting pose $(0,\ 0,\ 0°)$ and ending pose $(5,\ 4,\ 10°)$ can be seen. Thanks to equation 4.2 we can calculate tangent angle to any point at planned trajectory and therefore we can calculate the position of the rear motorized wheel $\mathcal{O}_\mathcal{W} = (x_w, y_w)$ at any s.

$$x_w(s) = x(s) - l_{wb} \cdot \cos(t(s)) \tag{4.5}$$
$$y_w(s) = y(s) - l_{wb} \cdot \sin(t(s)) \tag{4.6}$$

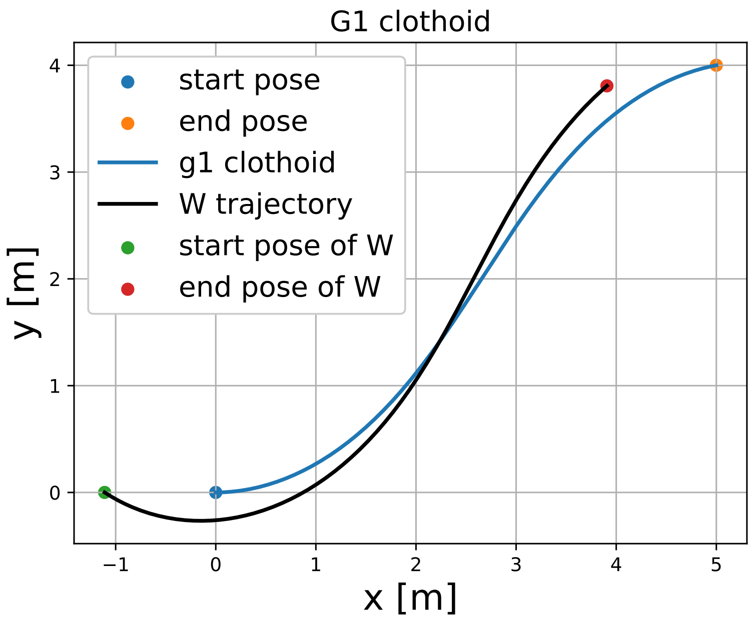


Figure 4.4: G^1 clothoid trajectories for $\mathcal{O}_\mathcal{F}$ and $\mathcal{O}_\mathcal{W}$.

Trajectory for the rear wheel alongside the clothoid trajectory is shown in figure 4.4.

As I have already mentioned before, G^1 clothoids do not allow us to plan trajectory with the starting curvature κ_0 which is unequivocally determined by the current steering angle γ of our vehicle. This is demonstrated in figure 4.5. This figure shows the steering angle parameterized by the arc-length s along the planned trajectory. To follow this path, we would need to turn the rear wheel to a certain angle before starting the motion. Then, we would proceed along the trajectory, and at the end, adjust the steering angle to zero again to make pallet loading possible.

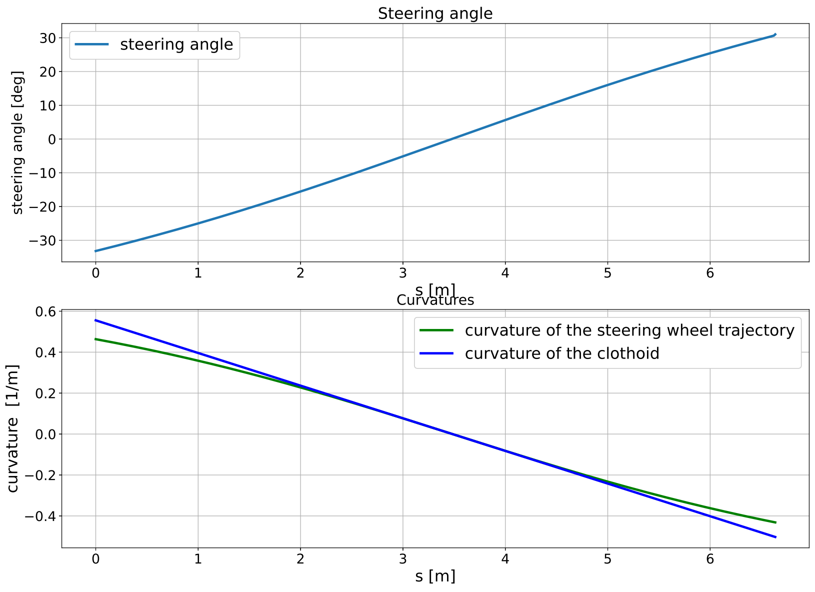
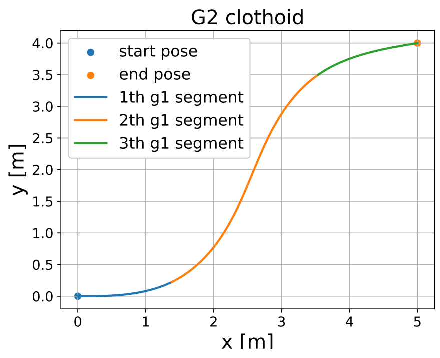


Figure 4.5: G^1 clothoid steering angles and curvatures.

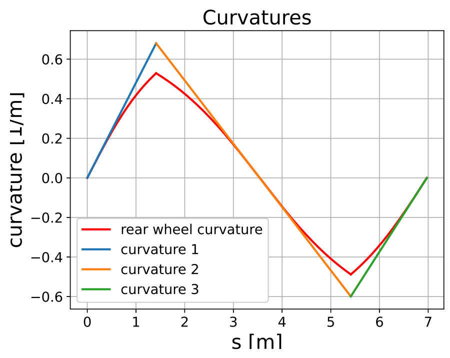
This approach is also unsuitable for visual servoing, where we try to estimate the pallet's position approximately three times per second, re-plan the trajectory accordingly, and drive along it until we receive another pallet pose estimation. To follow newly planned G^1 clothoid the pallet truck would have to adjust the steering angle accordingly and then proceed with the following. There is better alternative and it is called G^2 clothoid.

4.2.2 G^2 fitting with clothoids

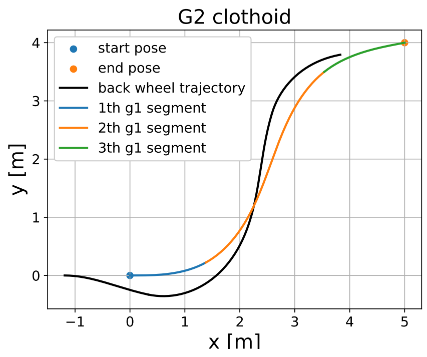
Inconveniences of G^1 clothoids can be solved by using G^2 Hermite Interpolation [6]. Using three clothoids connected to one trajectory we can satisfy the additional requirements for the starting and ending curvatures. Input for the G^2 Interpolation are six same parameters as in G^1 clothoid interpolation (x_0, y_0, t_0) and an ending pose (x_f, y_f, t_f) but we also add the requirements for the curvature at the start κ_0 and curvature at the end κ_f. The κ_0 is determined by the current steering angle γ and κ_f will be in our case equal to zero so ideally there is no need for steering adjustments after reaching the desired pose in front of the pallet.



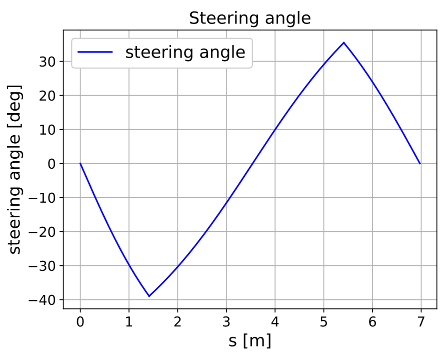
(a) : G^2 clothoid.



(b) : G^2 curvatures.



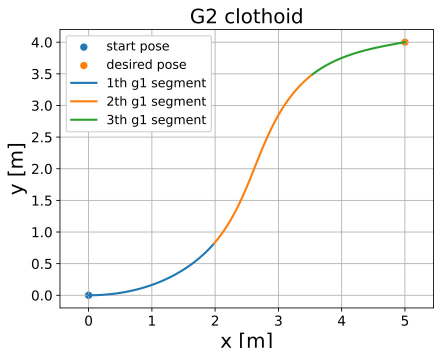
(c) : G^2 rear wheel trajectory.



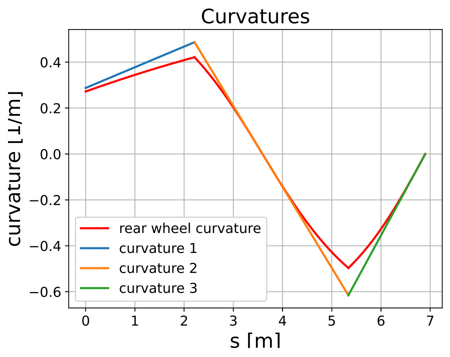
(d) : Steering angles along trajectory.

Figure 4.6: G^2 clothoid.

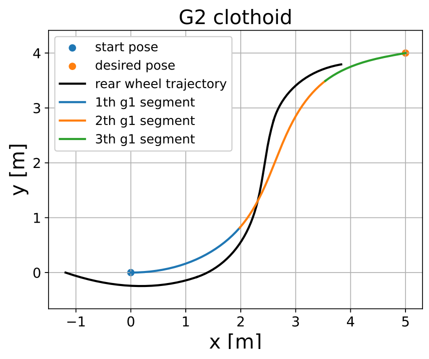
In figure 4.6 a G^2 clothoid trajectory with zero starting curvature κ_s is shown. Figure 4.7 shows a G^2 clothoid trajectory with a non-zero starting curvature κ_s, taking into account the current steering angle γ of the pallet truck.



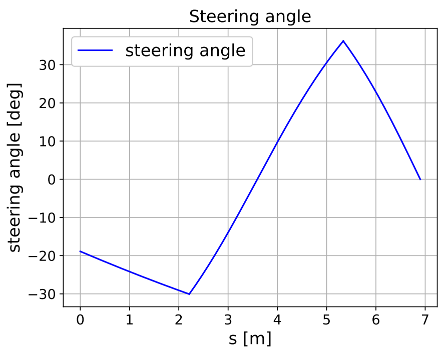
(a) : G^2 clothoid.



(b) : G^2 curvatures.



(c) : G^2 rear wheel trajectory.

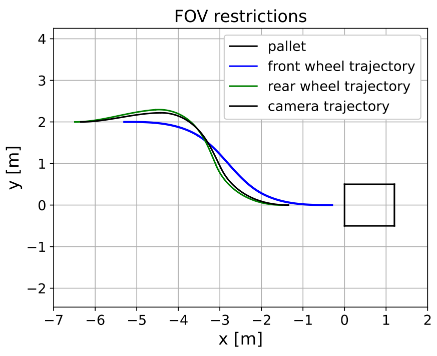


(d) : Steering angles along trajectory.

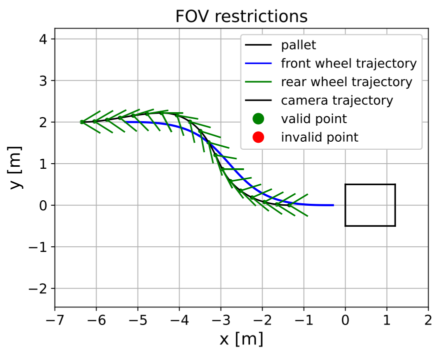
Figure 4.7: G^2 clothoid with a non zero starting curvature κ_s defined by the current steering angle γ of the pallet truck.

4.2.3 Visual servoing restrictions

A visual servoing approach requires that the pallet remains visible to the visual systems at all times during the ride. This means the pallet must stay within the field of view (FOV) of our RGBD camera. The camera's FOV is 69°. The graphs are generated for a FOV of 62° to account for a margin of error.

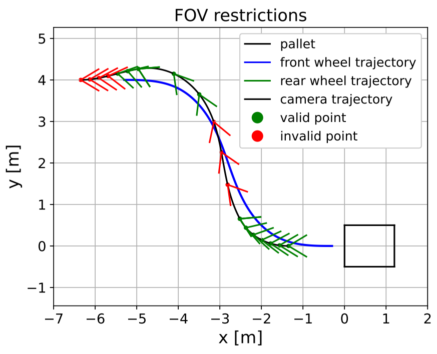


(a) : Frame's trajectories.

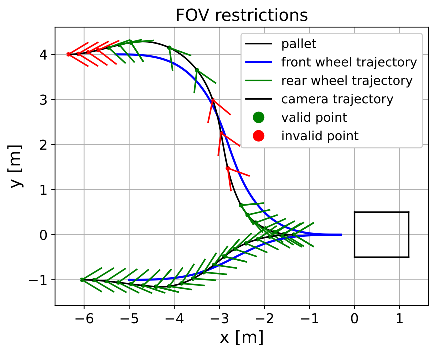


(b) : FOVs alongside trajectory.

Figure 4.8: FOV restrictions. Each frame on the vehicle $\mathcal{F}, \mathcal{C}$ and $\mathcal{W}$ follows slightly different trajectory. FOVs alongside trajectory are depicted via a green V-shaped boundaries. If the pallet is not on the FOV of the vehicle, the V-shape becomes red.



(a) : Invalid trajectory.

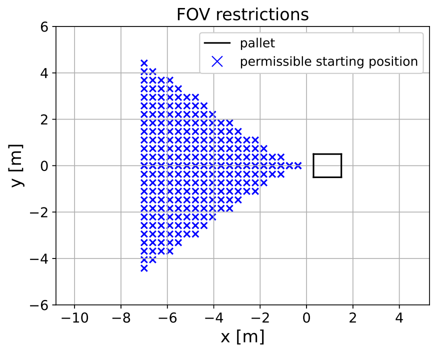


(b) : Comparison of valid/invalid trajectory.

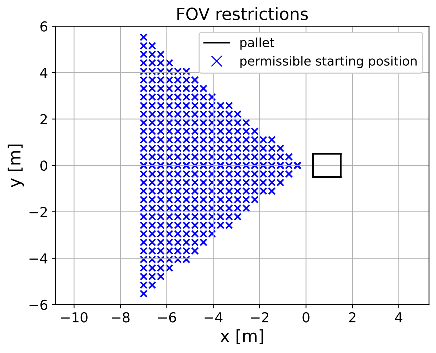
Figure 4.9: Invalid trajectories.

We need to determine which poses in front of the pallet satisfy the condition that a trajectory can be planned from them such that the camera sees the palette all the time. For the simplicity we assume that the curvature at the start is zero.

In figure 4.10 the blue crosses represent a position from which a valid clothoid, meeting the requirements for visual servoing, can be planned. Following figure 4.11 shows 3D graphs of the same situation only in first quadrant



(a) : Requirement to see the whole pallet during the ride.



(b) : Requirement to see the front side of the pallet during the ride.

Figure 4.10: Permissible starting positions in front of the pallet where the desired pose $\mathcal{O}_{\mathcal{D}}$ is based at (0, 0) with orientation 0°.

(due to the symmetry) and on thy y-axis are the permissible heading angles of the vehicle.

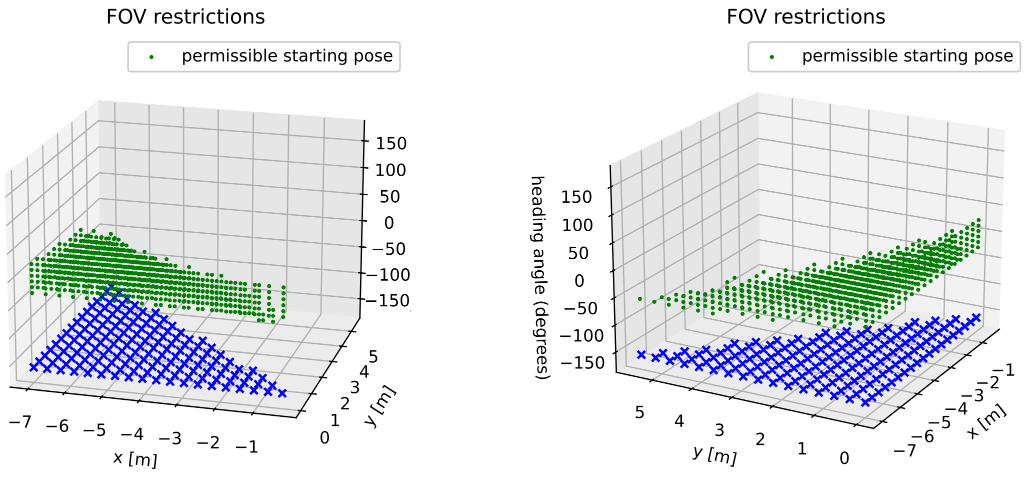


Figure 4.11: Permissible starting poses in front of the pallet, while the starting curvature κ_0 is zero and the desired pose $\mathcal{O}_{\mathcal{D}}$ is based at (0, 0) with orientation 0°.

Chapter 5

Control

5.1 Different sampling speeds of the control loops

The faster inner loop highlighted in figure 5.1 in blue works at sampling period $T_f = 32$ ms (in the simulation). The outer loop highlighted in yellow works at much lower speed with sampling period $T_s \sim$ 300 ms.

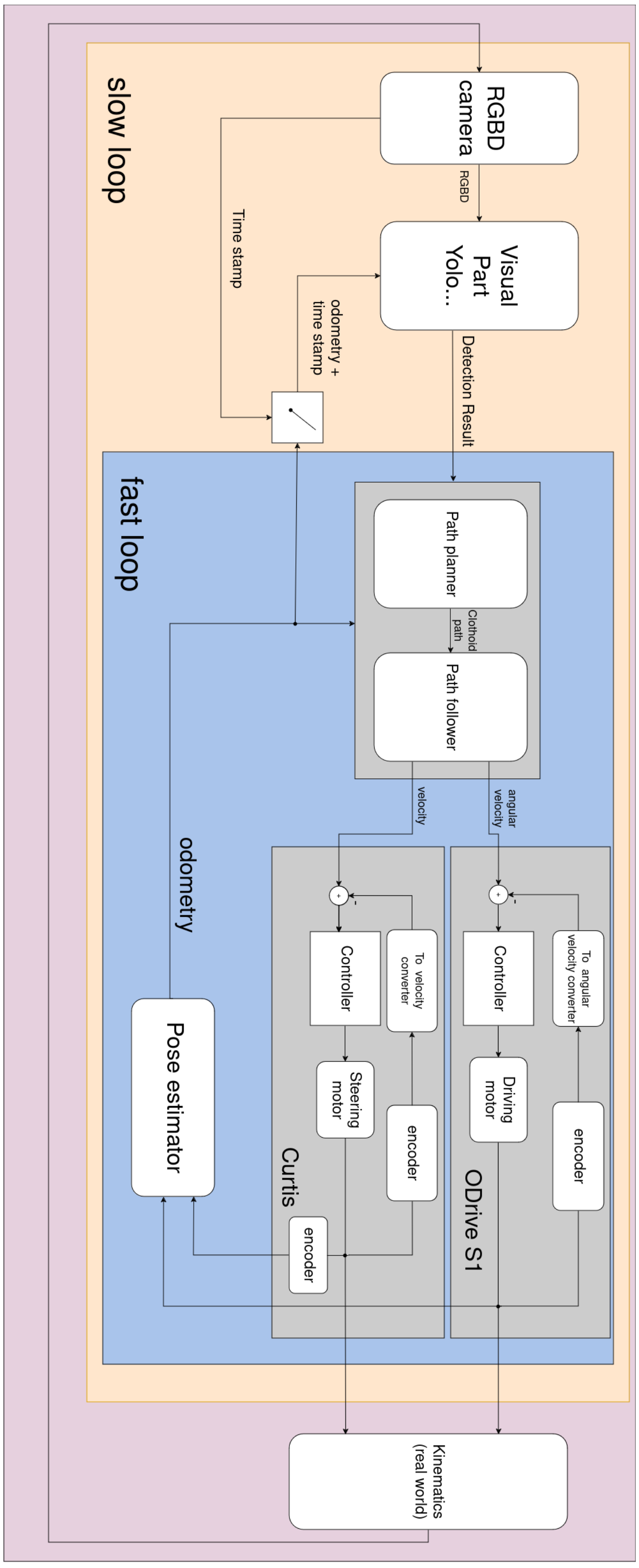


Figure 5.1: Control Scheme.

5.2 Dealing with the delay

A significant problem is that we will receive an estimated position of the pallet (detection result) with a non-negligible delay. We expect that processing the data will take about 700 ms (composition of the delay can be seen in figure 5.2.) This means that when we get the pallet pose estimate, it is already outdated by 700 ms, and we must account for this delay. This issue is discussed in [12].

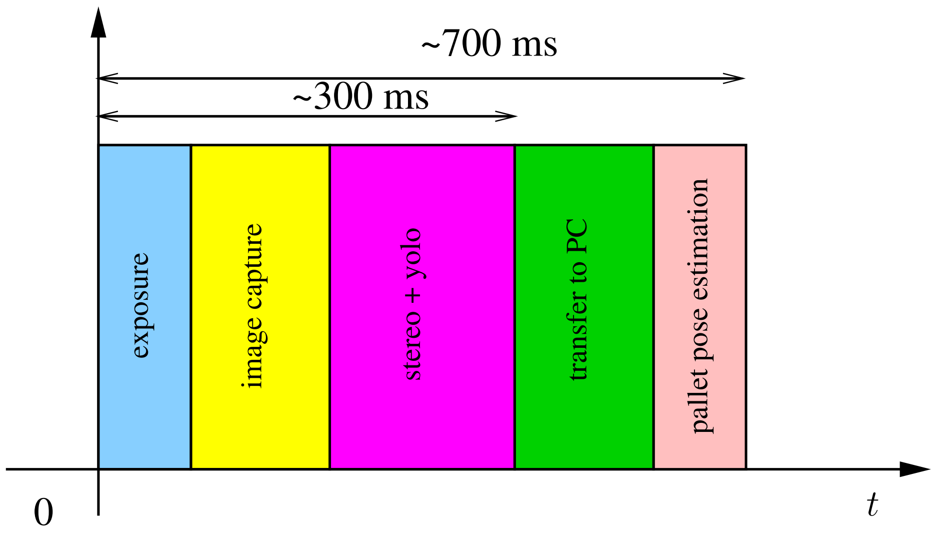


Figure 5.2: The delay between capturing an image and obtaining the pallet pose estimate is composed of several stages. During exposure, image capture, and stereo + YOLO processing (more details in [20]), the camera is occupied and cannot take another picture. Therefore, we aim to obtain new pallet pose estimates with a sampling period of approximately 300 ms. The entire process, from taking the picture to producing the pose estimate, is expected to take about 700 ms

If we do not take the delay into account, several problems arise as shown in figure 5.3. The cameras capture a picture of the scene, which then needs to be processed and transferred to the PC where the pallet pose estimation is performed. This process is illustrated in figure 5.2. During this image processing, the pallet truck continues to move ($\mathcal{C} \mapsto \mathcal{C}'$, $\mathcal{F} \mapsto \mathcal{F}'$), rendering the relative estimated pallet pose to the vehicle no longer valid.

If we plan the clothoid path based on the outdated estimated pallet pose, we will not reach the desired pallet position $\mathcal{D}$ but $\mathcal{D}'$. The solution is to integrate the motor speeds to get an estimate of the current pose of the vehicle once the old data arrives. After we receive the detection result from the vision system we have old pose of the pallet truck (pose when the image from which the pallet pose estimate is calculated was taken), the estimated pallet pose relative to the old pose of pallet truck and the current pose of the pallet truck. With these three information we can calculate an adjusted pallet pose estimation that is relative to the truck's current pose and plan a valid clothoid path accordingly.

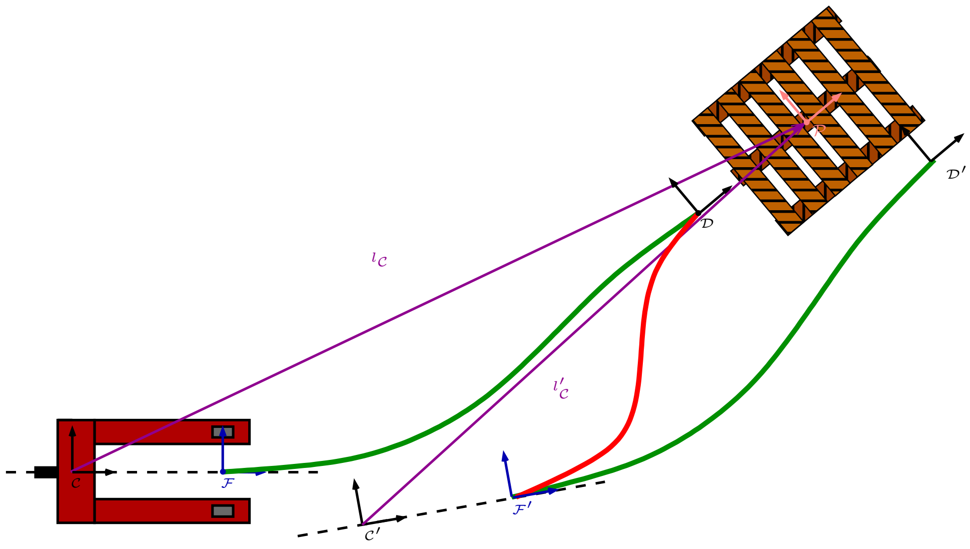


Figure 5.3: Delayed pallet pose estimate. The pallet pose estimate $(l_{\mathcal{C}}, \phi, \delta)$ is computed based on the image captured at the time when the pallet truck positions is at $(\mathcal{C}, \mathcal{F})$ Desired pose in front of the estimated pallet pose is $\mathcal{D}$. During the computation of the estimation (depicted in figure 5.2) the vehicle moves to a new pose $(\mathcal{C}', \mathcal{F}')$ and the relative estimated position of the pallet to the vehicle and the desired pose $\mathcal{D}'$ are no longer valid. Without the correction we would follow an incorrectly planned path.

5.3 ROS scheme

We are using ROS2 [14] for easy communication between nodes and possible replacements of individual nodes with a different ones. It is also a technology which KM robotics [21] uses. Scheme of our ROS communication can be seen in figure 5.4.

5.4 Pure Pursuit Controller

Pure Pursuit is a geometry based controller [22], [1]. It's parameter is the look ahead distance l_d and it calculates necessary steering angle to reach a point on the path in the look ahead distance.

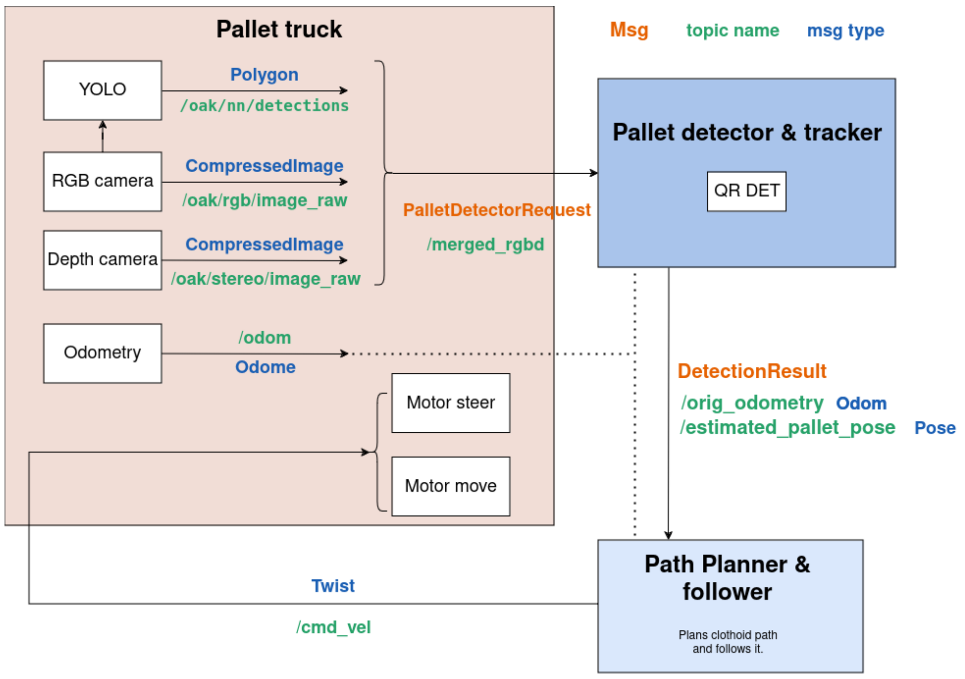


Figure 5.4: ROS communication scheme.

Firstly the look-ahead point (LP) is detected. It lies on the intersection of the circle with the center at origin of $\mathcal{F}$ ($\mathcal{O}_\mathcal{F}$) with radius l_d and the planned path. To reach this point LP, the steering angle γ has to be adjusted so the frame $\mathcal{F}$ would follow circular path that would intersect with this point. Since the look-ahead point must lie on the circle denoting the circular path of the $\mathcal{F}$ we can calculate needed steering angle to reach this point LP. Angle α is the angle between the axis of the vehicle and the line connecting $\mathcal{F}$ and LP. The triangle formed by the points ICR, $\mathcal{O}_\mathcal{F}$ and LP is isosceles and therefore $\beta_2 = \beta_3$. The relation between β_1 and α. is:

$$180^\circ = \beta_1 + \beta_2 + \beta_3 = \beta_1 + (90^\circ - \alpha) + (90^\circ - \alpha)\,. \tag{5.1}$$

This means that $\beta_1 = 2\alpha$. The law of sines tells us following:

$$\frac{l_d}{\sin \beta_1} = \frac{r_\mathcal{F}}{\sin \beta_2}\,. \tag{5.2}$$

After substituting in the values for $\beta_1 = 2\alpha$ and $\beta_2 = 90^\circ - \alpha$:

$$\frac{l_d}{\sin(2\alpha)} = \frac{r_\mathcal{F}}{\sin(90^\circ - \alpha)}\,, \tag{5.3}$$

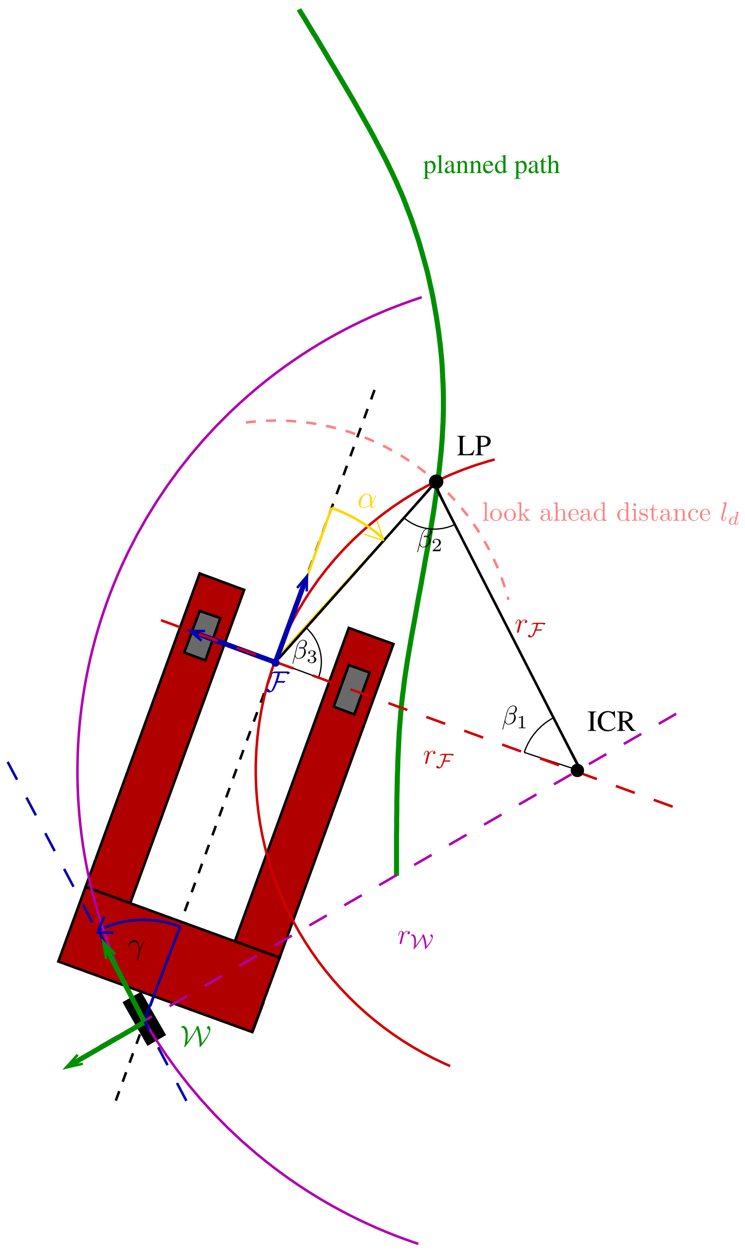


Figure 5.5: PPC computes needed steering angle γ to reach look-ahead point LP in the l_d distance from $\mathcal{O}_{\mathcal{F}}$. α is the angle between the axis of the vehicle and the line connecting $\mathcal{O}_{\mathcal{F}}$ and LP. Knowing the constant l_{wb} we can calculate the desired steering angle γ_d as shown in equation 5.5.

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{r_{\mathcal{F}}}{\cos(\alpha)}\,, \tag{5.4}$$

the calculated $r_{\mathcal{F}} = \frac{l_d}{2\sin(\alpha)}$ corresponds to the desired steering angle γ_d

which can be calculated based on the relation mentioned before 3.11:

$$\gamma_d = \arctan\left(\frac{2l_{wb} \cdot \sin(\alpha)}{l_d}\right) . \tag{5.5}$$

Chapter 6

Simulation

6.1 Setup

We were using Webots (an open-source mobile robot simulation software [25]) for the simulations. A lot of work regarding setting up the simulation had been already done by the Miroslav Uller and Japanese students (Ayne Hokari and Ryohe Tozaki). Thanks to them, the initial setup of the simulation was already implemented. We can add several pallets and our pallet truck model to the simulation scene.

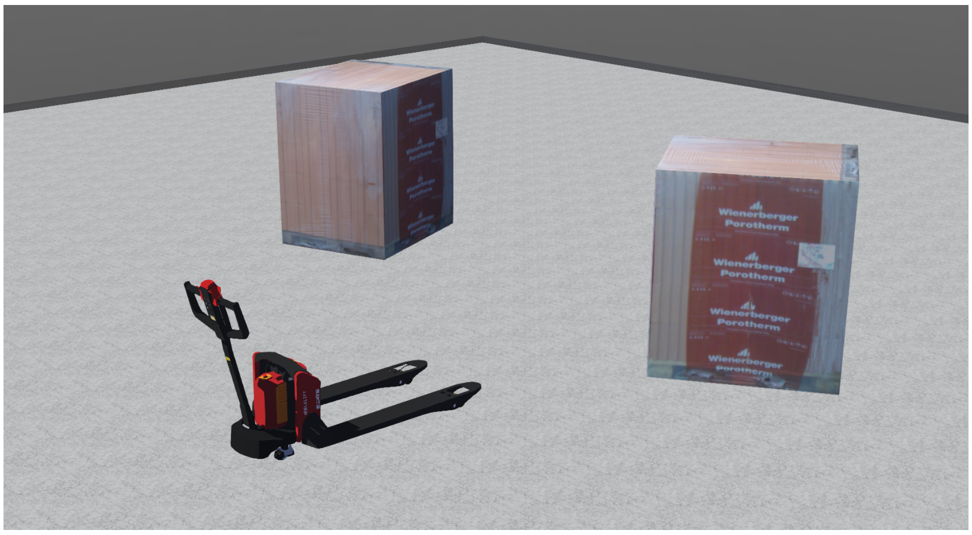
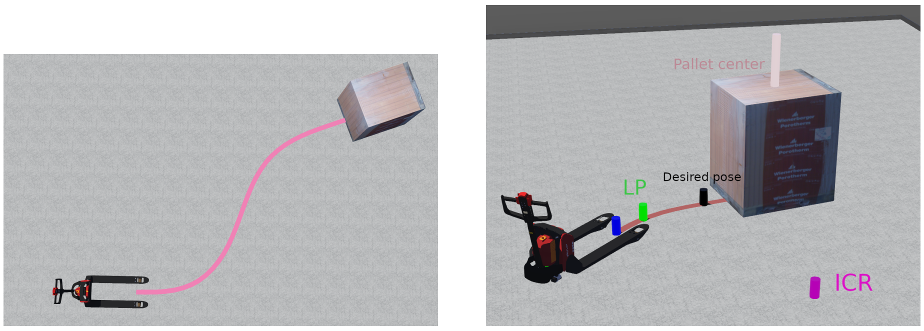


Figure 6.1: Simulation scene setup.

Whenever a visual system sends a pallet estimated position, we can visualize planned trajectory to detected pallet. Furthermore we can display several markers such as estimated pallet pose $\mathcal{P}$ target pose $\mathcal{D}$ look-ahead point LP and ICR.



(a) : Trajectory visualization. **(b) :** Markers visualization.

Figure 6.2: Simulation visualization.

At the moment the camera and the depth sensor are mounted on the simulated pallet truck, and can their data can be published via ROS topics however the visual part is not fully implemented and integrated in to the simulation.

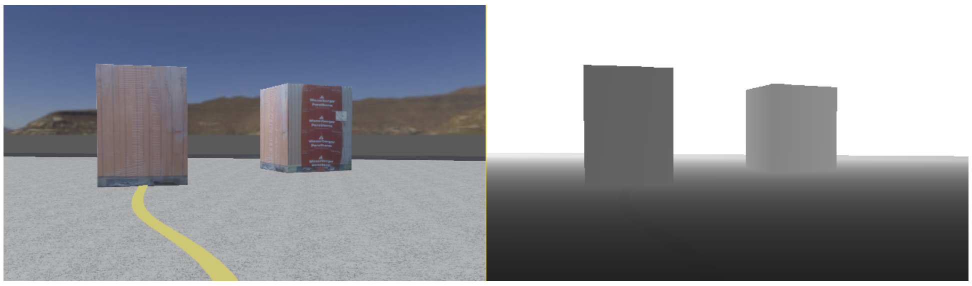


Figure 6.3: Simulated camera and depth sensor.

6.2 Simulation operation

From the simulator we have an easy access to all important parameters such as current position of the pallet truck model, its steering angle motor speeds and their position. We can also access all the information about the pallets in the scene.

To address the absence of a visual system, we are introducing a "fake detector node" as depicted in figure 6.4. This node subscribes to the */odom* and */pallet_pose* topics provided by the simulator. It mimics the behavior of the visual system being developed by Richard Randák.

When the node is not busy processing data from the sensors, it records the last known odometry. It then simulates the data processing by waiting for a specified amount of time. After this period, the node publishes a DetectionResult message, which includes the last recorded odometry and the relative pallet pose to the odometry.

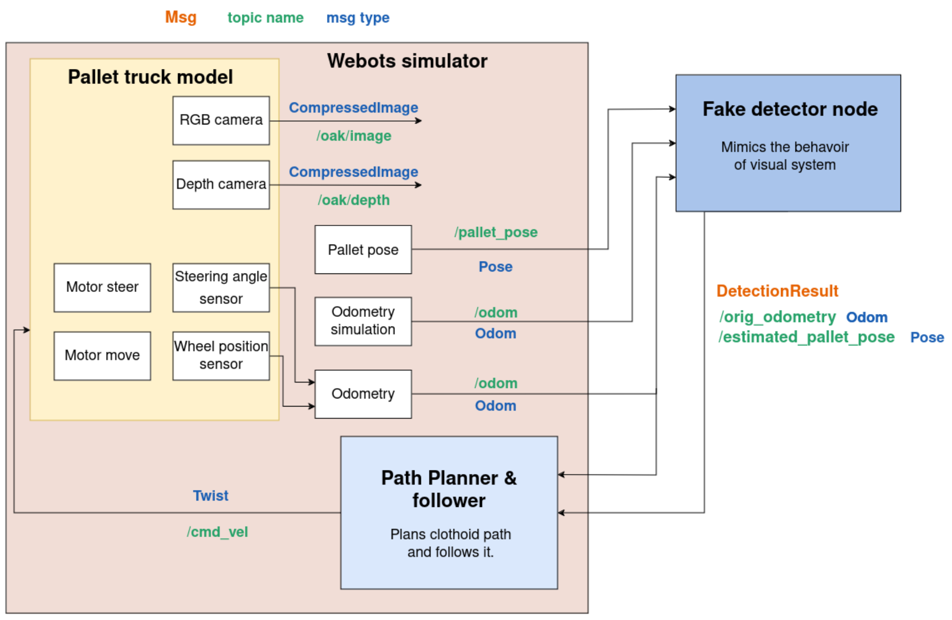


Figure 6.4: ROS communication in simulation. The difference from the ROS Communion scheme in figure 5.4 is that in simulation we are using the fake detector node instead of the actual node (developed by Richard Randák in [20]) which takes care of the image processing and detection of the pallet. The Path planner and follower node is in the environment of the webots [25].

The path planner subscribes to the */odom* topic and receives the Detection-Result message. Based on the current position of the pallet truck it adjusts the relative pallet pose and plans a G^2 clothoid trajectory. The PPC then follows the clothoid until the new pallet pose estimate arrives and the process repeats. The PPC works at the sampling period $T_f = 32$ ms. In each loop PPC calculates desired steering angle γ_d and because we are using motor speed control we calculated desired angular velocity:

$$\omega_d = \frac{\gamma_d}{T_f}\,. \tag{6.1}$$

6.2.1 Odometry calculation

Odometry is initialized at initial pose, (x_0, y_0, ϕ_0) and current steering angle γ of the pallet truck model. Each simulation step when the motors move (steering angle and wheel position changes) The change in wheel position $\Delta\varepsilon$

is calculated if there is a previous wheel position. This change is used to update the pose.

The distance traveled is:

$$d = \Delta\varepsilon \cdot r_d \,. \tag{6.2}$$

If the steering angle γ is very small, indicating straight movement, the pose updates are:

$$\Delta x = d\cos(\phi)\,, \tag{6.3}$$

$$\Delta y = d\sin(\phi)\,. \tag{6.4}$$

For non-zero steering angles, the radius of the path ($r_{\mathcal{F}}$) and the wheel path radius ($r_{\mathcal{W}}$) are calculated as:

$$r_{\mathcal{F}} = \frac{l_{wb}}{\tan(\gamma)}\,, \tag{6.5}$$

$$r_{\mathcal{W}} = \frac{l_{wb}}{\sin(\gamma)}\,. \tag{6.6}$$

The angle of arc movement $\Delta\phi$ is:

$$\Delta\phi = \frac{d}{r_{\mathcal{W}}}\,. \tag{6.7}$$

The relative changes in position are:

$$\Delta x_r = r_{\mathcal{F}}\sin(\Delta\phi)\,, \tag{6.8}$$

$$\Delta y_r = r_{\mathcal{F}}(1-\cos(\Delta\phi))\,. \tag{6.9}$$

These changes are rotated to align with the current orientation:

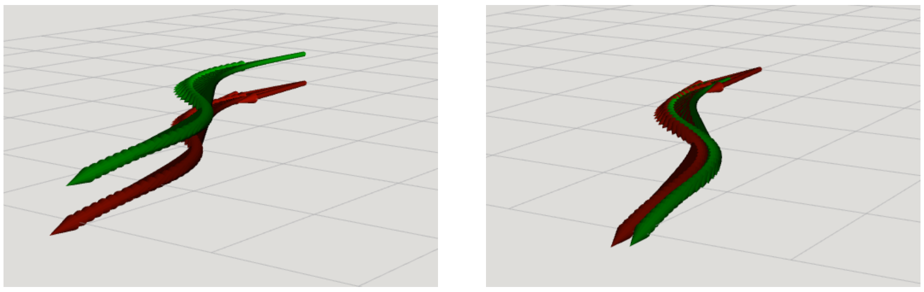
$$\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} = \boldsymbol{R(\phi)}\begin{pmatrix}\Delta x_r\\ \Delta y_r\end{pmatrix}\,. \tag{6.10}$$

The orientation is updated as follows:

$$\phi = \phi + \Delta\phi\,. \tag{6.11}$$

The steering angle is updated with the current steering angle from the sensor.

In the simulator, we have access to the actual pose of the pallet truck model. In figure 6.5, the comparison between the calculated odometry and the actual odometry from the simulator can be seen. The red arrows represent the odometry from the simulator, and the green arrows represent the calculated odometry based on the mounted sensors. In figure 6.5a, the odometries are offset along the z-axis for better visualization. In figure 6.5b, a slight error between them can be seen, as the calculated odometry does not represent the motion of the pallet truck exactly accurately.



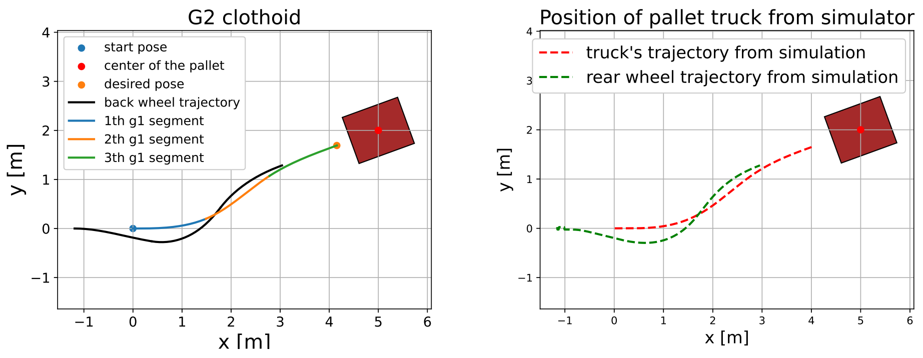
(a) : Odometries **(b) :** Odometries

Figure 6.5: Rviz comparison of odometries. Red arrows represent pose of the pallet truck from the simulator and the green arrows represent poses calculated based on the sensors in the pallet truck. In figure 6.5a, the odometries are offset along the z-axis for better visualization. We can see, that these 2 odometries slightly differ and they drift a bit from each other over time.

6.3 Simulation results

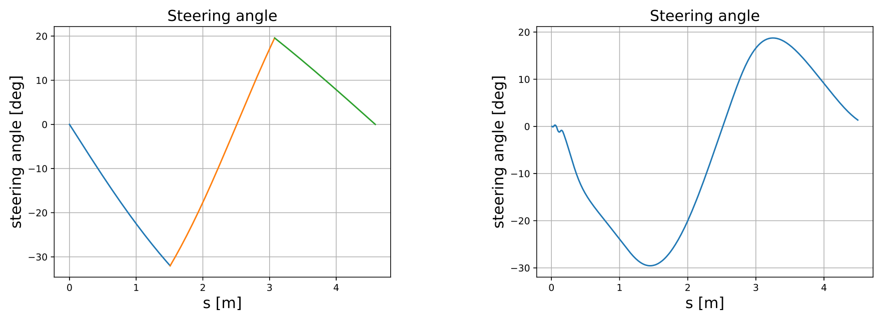
6.3.1 Simplest case - following single G2 clothoid

Considering the simplest case, when we want to follow pre-planned trajectory consisting of three clothoids. Trajectory is not updated during the ride and pallet pose estimate is accurate.



(a) : Planed trajectories. **(b) :** Simulated trajectories.

Figure 6.6: Single G^2 clothoid.



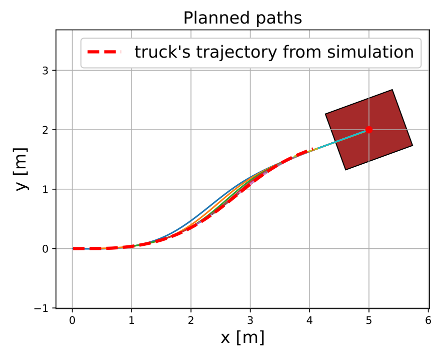
(a) : Theoretical steering angles. **(b) :** Steering angles from the simulator.

Figure 6.7: Simplest case steering angles comparison.

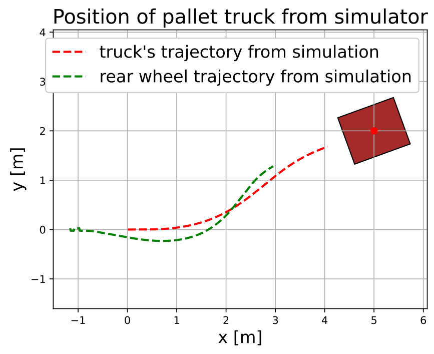
It is not realistic to achieve the theoretical steering angles shown in figure 6.15a. The derivation is not a continuous function, and the steering motor is controlled by continuous current.

6.3.2 Planned path updates in low frequency

When we implement the visual servoing approach, the new planned path will slightly differ from the previously planned path.

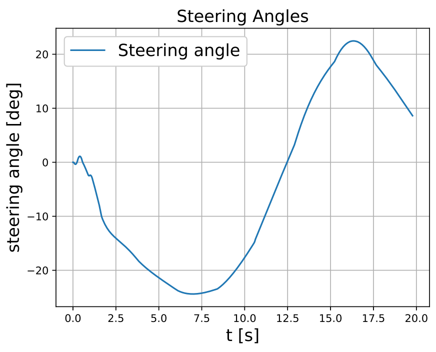


(a) : Planed trajectories.

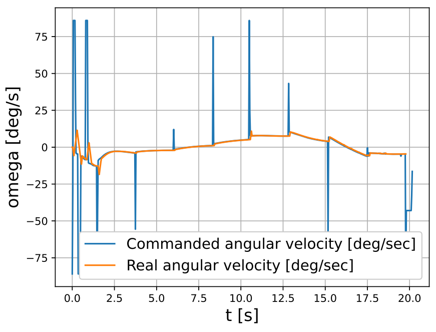


(b) : Simulated trajectories.

Figure 6.8: Multiple trajectories.



(a) : Steering angles from the simulation.



(b) : ω for the steering wheel.

Figure 6.9: Low frequency trajectory updates.

6.3.3 Considering noisy pallet pose estimate

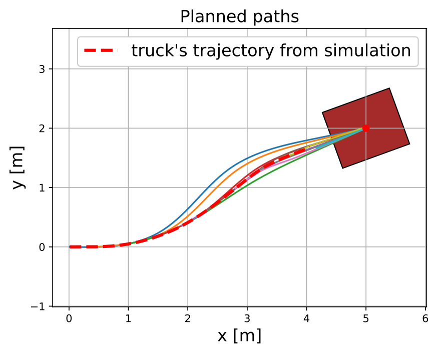
Since the pallet pose estimates will not be accurate we expect the pallets estimated normal angle to be slightly different each time. We are adding this error manually in the simulation by adding some noise to the normal angle ϕ of the pallet. The closer we are to the pallet the estimates should get better. We are modelling the noise as follows:

$$f \sim \mathcal{U}(-1, 1)\,, \tag{6.12}$$

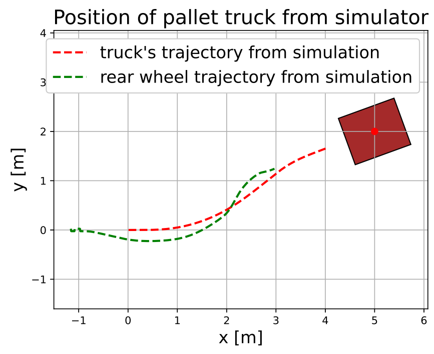
$$\phi_{max}(l) = \frac{l}{35}\,, \tag{6.13}$$

$$\phi' = \phi + \phi_{max}(l) \cdot r\,, \tag{6.14}$$

where f is random factor, l is the distance of the vehicle from the pallet and $\phi_{max}(l)$ is maximal deviation in the ϕ as function of l.

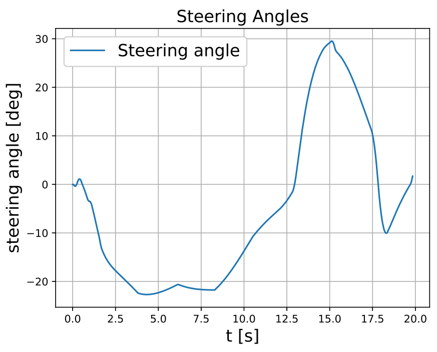


(a) : Planed trajectories.

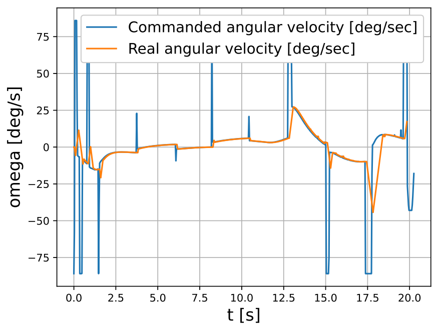


(b) : Simulated trajectories.

Figure 6.10: Trajectories with noisy pallet estimates.



(a) : Steering angles from the simulation.

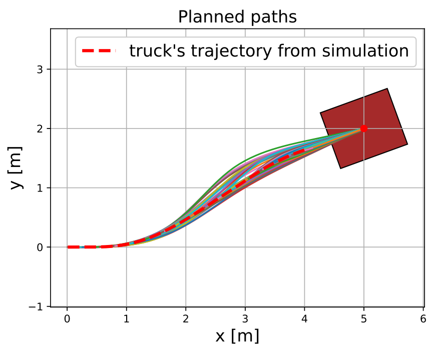


(b) : ω for the steering wheel.

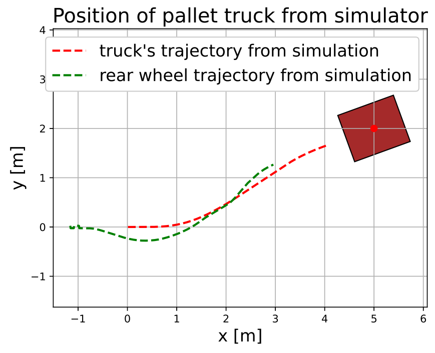
Figure 6.11: Low frequency trajectory updates considering noisy pallet pose estimates.

6.3.4 Planned path updates in high frequency

The updates of the estimated pallet pose should come with higher frequency. This case is depicted in figures 6.12 and 6.13.

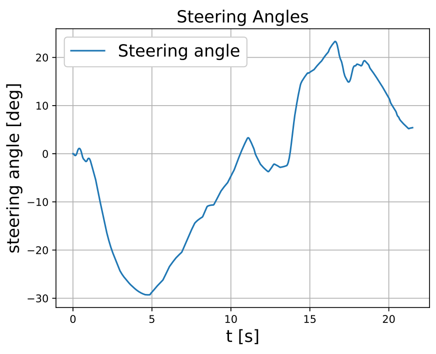


(a) : Planed trajectories.

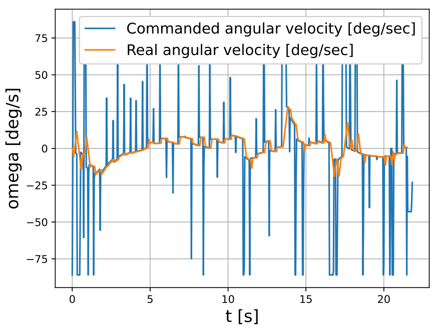


(b) : Trajectories from the simulator.

Figure 6.12: High frequency updated trajectories with noisy pallet estimates.



(a) : Steering angles from the simulation



(b) : ω for the steering wheel.

Figure 6.13: High frequency trajectory updates considering noisy pallet pose estimates.

6.3.5 Rate of change in angular velocity is too high

The PPC calculates desired steering angle for the vehicle. We are controlling the steering motor with the velocity control. Therefore we need to get the ω_s from the desired steering angle γ_d. If the control loop runs at frequency f_{cl} and its period is T_f we can calculate ω_s as follows:

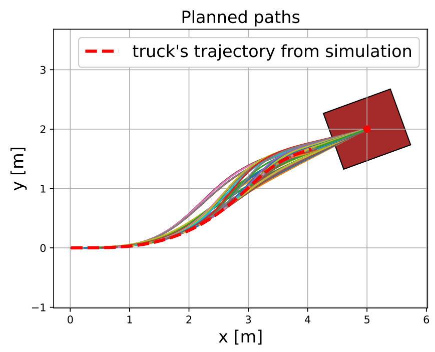
$$\omega_s = \frac{\gamma_d}{T_f} \tag{6.15}$$

This results in high peaks in the velocity control for the steering motor when the new pallet pose estimate arrives.

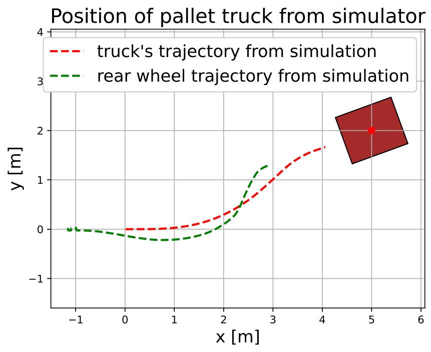
We can filter the the desired angle γ_d calculated by the PPC:

$$\gamma_{d\ filtered} = k \cdot \gamma_d + (1-k) \cdot \gamma\,, \tag{6.16}$$

where γ is the current steering angle of the pallet truck and k is the filter constant. For this simulations the was set to $k = 0,05$.

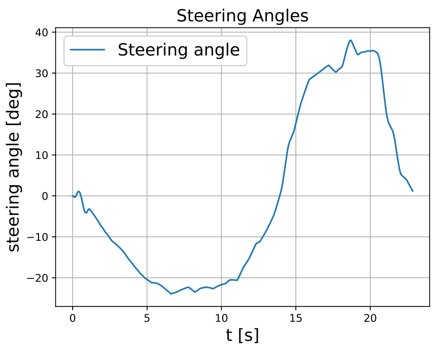


(a) : Planed trajectories.

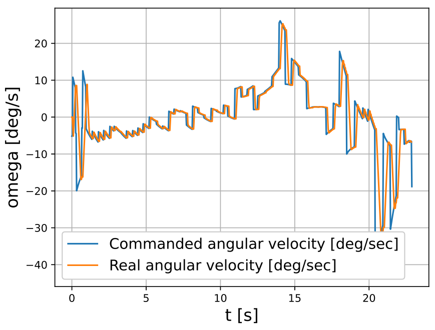


(b) : Trajectories from the simulator.

Figure 6.14: High frequency updated trajectories with noisy pallet estimates.



(a) : Steering angles from the simulator



(b) : ω for the steering wheel.

Figure 6.15: Control inputs with filtered γ_d.

Chapter 7

Conclusion

In this bachelor thesis, our objective was to design trajectories for a pallet truck and develop its control system. We determined that utilizing the Visual Servoing approach is essential because the estimation of the pallet position lacks accuracy at higher distances and the accuracy only improves when the pallet is in closer proximity. We conducted a detailed examination of the properties of clothoid curves as potential paths for the pallet truck and concluded that they are suitable for this purpose. Then the Pure Pursuit Control (PPC) algorithm was implemented for the trajectory tracking. Through several experiments it has proven it's capability to accurately track the trajectory even with frequent re-planning necessitated by the Visual Servoing approach.

Due to time constraints, we were unable to test our design on a physical model of the pallet truck. All testing was conducted within a simulation environment. Despite this limitation, the implementation in the simulation environment, facilitated by the Robot Operating System (ROS2), ensures that the transition from simulation to real-world experiments should be seamless.

Appendix A

Software libraries

- Numpy
  This library is used for mathematical operations and batch operations on data. We used it in version 1.26.2. More information alongside the library's documentation is available at [2]
- Shapely
  This library is used to work with geometric objects. We used it in version 2.0.3. Documentation for this library is available at [9]
- Matplotlib
  This library was used to generate all the graphs in this thesis. [7]. We used the version 3.8.3.
- Pyclothoid
  This library is used for the clothoid computation [8]. We used version 0.1.4.

Developed Code

All of the developed code is available on GitHub:

- Clothoid graphing code:
  `https://github.com/Adabas01/clothoids.git`
- Pallet Truck Simulation Code:
  `https://github.com/mykkro/rovozci-dev.git`

Appendix B

Used AI and other supportive tools.

- Grammarly [11]
  Grammarly was used for some corrections in the text regarding grammar.
- ChatGPT [16]
  ChatGPT was utilized to enhance the visual presentation of graphs and to refine the wording of certain sentences for improved clarity.
- DeepL [17]
  DeepL was employed to translate certain words or phrases from my native Czech language to English.

Appendix C

Bibliography

[1] Mario theers ppc github. `https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Control/PurePursuit.html`, 2022. [Online].

[2] Numpy documentation. `https://numpy.org/doc/stable/`, 2022. Version 1.24 [Online].

[3] F. Chaumette A. Cherubini and G. Oriolo. An image-based visual servoing scheme for following paths with nonholonomic mobile robots. *Int. Conf. on Control, Automation, Robotics and Vision*, pages 108–113, 2008.

[4] F. Chaumette A. Cherubini and G. Oriolo. A positionbased visual servoing scheme for following paths with nonholonomic mobile robots. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1648–1654, 2008.

[5] Enrico Bertolazzi and Marco Frego. G1 fitting with clothoids. *Mathematical Methods in the Applied Sciences*, 38(5):881–897, 2015.

[6] Enrico Bertolazzi and Marco Frego. On the g2 hermite interpolation problem with clothoids. *Journal of Computational and Applied Mathematics*, 341:99–116, 2018.

[7] NumPy developers. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[8] Phillip Dix. pyclothoids, 2020. `https://pyclothoids.readthedocs.io/en/latest/about.html`, Last accesed on 2024-5-24.

[9] Sean Gillies. The shapely user manual. `https://shapely.readthedocs.io/en/stable/manual.html`, 2023. Version 2.0.1 [Online].

[10] Vicent Girbés, Leopoldo Armesto, and Josep Tornero. Path following hybrid control for vehicle stability applied to industrial forklifts. *Robotics and Autonomous Systems*, 62(6):910–922, 2014.

[11] Grammarly. `https://www.grammarly.com/`, Last accesed on 2024-5-24.

[12] Zdeněk Hurák and Martin Řezáč. Delay compensation in a dual-rate cascade visual servomechanism. *49th IEEE Conference on Decision and Control (CDC)*, pages 1639–1643, 2010.

[13] F Künhe, J Gomes, and W Fetter. Mobile robot trajectory tracking using model predictive control. In *II IEEE latin-american robotics symposium*, volume 51, 2005.

[14] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[15] Alireza Mohammadi, Iven Mareels, and Denny Oetomo. Model predictive motion control of autonomous forklift vehicles with dynamics balance constraint. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6, 2016.

[16] NOBLELIFT. `https://openai.com/index/chatgpt/`, Last accesed on 2024-5-24.

[17] NOBLELIFT. `https://www.deepl.com/translator`, Last accesed on 2024-5-24.

[18] NOBLELIFT, 2024. `http://noblelift.com/Material-Handling/Pallet-Truck/EDGE-PTE20N`, Last accesed on 2024-5-24.

[19] Odrive, 2024. `https://odriverobotics.com/shop/odrive-s1`, Last accesed on 2024-5-24.

[20] Richard Randák. Estimating the pose of a pallet with bricks using sensors of a pallet truck, 2024. bachelor's thesis, CTU Prague.

[21] KM robotics, 2024. `https://www.km-robotics.cz/R`, Last accesed on 2024-5-24.

[22] Mohammad Rokonuzzaman, Navid Mohajer, Saeid Nahavandi, and Shady Mohamed. Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intelligent Transport Systems*, 15, 03 2021.

[23] Tua Tamba, Bonghee Hong, and Keum-Shik Hong. A path following control of an unmanned autonomous forklift. *International Journal of Control, Automation and Systems*, 7:113–122, 02 2009.

[24] Rui Wang, Ying Li, Jiahao Fan, Tan Wang, and Xuetao Chen. A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller. *IEEE Access*, 8:166525–166540, 2020.

[25] Cyberbotics Ltd. Webots. http://www.cyberbotics.com. Open-source Mobile Robot Simulation Software.