Pedestrian Tracking and Collision Avoidance of the Mobile Robot

Master Project
Jieming Chen

Fachbereich Elektrotechnik und Informationstechnik
Lehrstuhl für Regelungssysteme

Betreuer:
Prof. Dr.-Ing. Steven Liu
Msc. Xiang Chen

Technische Universität Kaiserslautern

November 15, 2020

Erklärung

Die vorliegende Arbeit habe ich auf Initiative und unter Anleitung meiner Betreuer angefertigt. Bei der Erstellung habe ich keine anderen als die angegebenen Hilfsmittel verwendet.

Kaiserslautern, November 15, 2020

Musterstudent Jieming Chen

Contents

List of Figures

List of Tables

Nomenklatur

$\boldsymbol{C}$ Coriolis- und Zentrifugalmatrix

$\boldsymbol{M}$ Massenmatrix

1 Introduction

Autonomous vehicles are becoming a reality. Many companies, Baidu, ARGO AI, are working on developing self-driving cars and hardware for autonomous vehicles of all types. Interaction between vehicles and pedestrians is an important topic and still needs further research. It involves perception, trajectory planning, control, and many aspects.

The development in the fields of sensors and control that contribute to the autonomous vehicle technology has grown rapidly in the past decade. Sensor technology is important with the perception component. Perception has outstanding progress due to the deep neural network.

Another important component, control has advanced to bridge the gap between lower-level control towards higher-level planning stages.The planning and control systems of autonomous driving are responsible for generating a safe trajectory and following this trajectory, considering the dynamic environment and nonholonomic vehicles. Perception, Planning, control topics are explored in this master project to develop a whole framework for interaction between people and vehicles.

1.1 Aim and motivation

Safety is the top topic for both people and autonomous vehicles, so collision avoidance needs to be guaranteed in real time. Intelligence is an another factor. Vehicles should consider behaviour of people, before moving. The aim of this project is to implement a framework that could avoid pedestrian from perception to control.

Due to the great performance of DNN, the perception part considered to be implemented by DNN.

In the motion planning part, searching methods, A*, RRT could not take behaviour of people into account. Optimization based prediction control is suitable for this kind of task. For the precise control, model based method is taken. Therefore, model predictive control is implemented to generate trajectories and control a vehicle.

1.2 State of the art methods

There has been much work in developing object detection algorithms using a standard camera with no additional sensors. State-of-the-art object detection algorithms use deep neural networks. Convolutional Neural Networks (CNNs) is the main architecture that is used for computer vision. Instead of having fully-connected layers, a CNN has a convolution layer where a filter is convolved with different parts of the input to create the output. [RF18] is a fast and accurate CNN model. Based on the convolutional neural network (CNN) based detector, the [BGO+16] presents a lean implementation of a tracking-by-detection framework for the problem of multiple object tracking (MOT) where objects are detected each frame and represented as bounding boxes.

Classical approaches to obstacle avoidance include [KB+91]. These approaches do neither produce optimal trajectories, nor unify planning and control, nor account for complex robot dynamics. A simple and effective method that is still used in practice, is the elastic-band algorithm [QK93]. The computed paths, however, are generally non-smooth, i.e. they are not guaranteed to satisfy kinodynamic constraints, nor does this algorithm compute a velocity profile. In [SPAD19], trajectory optimization methods try to find time-optimal and collision-free robot trajectories by formulating and solving an optimization problem.

In robotics, the generation of trajectories with avoidance constraints has been extensively studied. According to [SNS11], two different approaches exist: planning and reacting. The planned approach generates feasible paths ahead of time; whereas the reactive approach typically uses an online collision avoidance system to respond to dangerous situations. [AMA14] proposed many methods to convert non-convex constriants to convex constriants.

1.3 Research goals

The two main research goals were perception of people and controller design, but the focus area of this study was dominated by the model predictive controller development. The objective to enable the vehicle to operate autonomously to perform collision avoidance in static and dynamic situations. The following way points guided the project work.

- People tracking based on CNN, Kalman filter and a data association method.
- Formulation of the trajectory generation probleam as a QP problem.
- Evaluate the performance of the controller in the simulation and real vehicle.

1.4 Organisation of the report

The following report is structured to introduce the tracking algorithm, controller development methods, and the experimental results.

Chapter 1 is used to introduce the report focus areas of autonomous vehicle technology. The aim and motivation of the topics are described and a brief summary of the state of the art methods leads to research goals.

Chapter 2 details the kinematic model of the vehicle. The bicycle model is described and the states and control variables are clearly established.

Chapter 3 introduces sensors, detection algorithm, tracking algorithm for the pedestrian tracking.

Chapter 4 is used to elaborate the controller design. The optimization problem is formulated and approximation to QP problem is detailed.

Chapter 5 analysis the performance of the controller in static and dynamic environment.

Chapter 6 concludes the report and recommendations of future work.

2 System model

This chapter details the vehicle kinematic model with respect to the bicycle model used for controller development.

2.1 Coordinate system

The Cartesian co-ordinate system is used as the reference frame. The motion of the vehicle can be described using the X-Y co-ordinates. Here, the longitudinal motion of the vehicle is along the X-axis and the lateral motion of the car is along the Y- axis. This representation will be followed throughout the rest of the thesis.

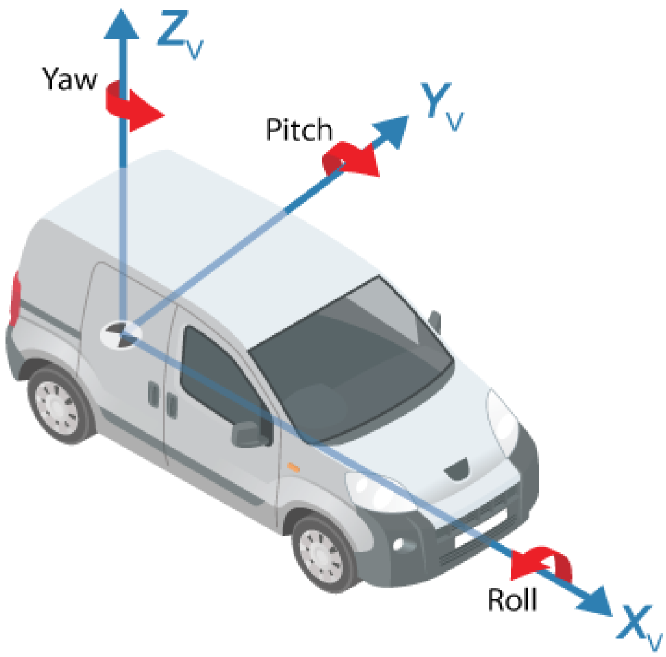


Figure 2.1: Co-ordinate system of the vehicle defining the longitudinal and lateral motion of the vehicle

The advantage with the Cartesian coordinate system is seen with the ability to capture the position and orientation essentially, denoting the all the available time derivatives. Any vector can be assigned a space using the three axes system of the coordinate system. Reference frames are helpful to understand the movements of vectors relative to each other.

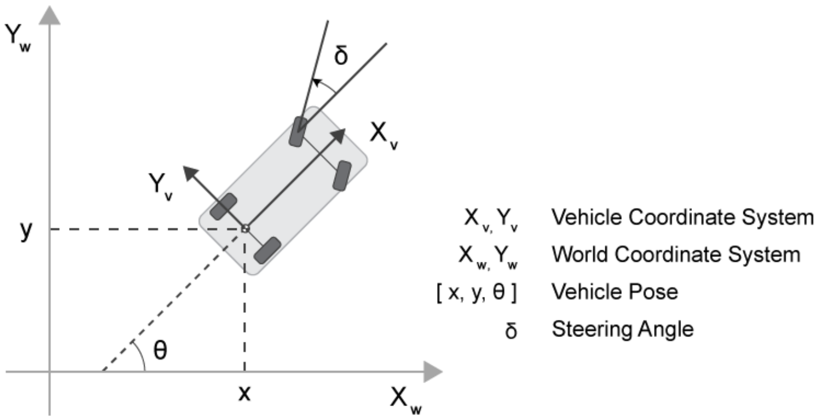


Figure 2.2: Kinematic bicycle model

Our case is the low-speed condition, so we only need to consider the kinematic model.

$$
\begin{aligned}
\dot{x} &= v * cos(\theta) \\
\dot{y} &= v * sin(\theta) \\
\dot{\theta} &= v * tan(\delta)/L
\end{aligned}
\tag{2.1}
$$

where x represents the position of the vehicle model in the longitudinal direction, y represents the position of the vehicle model in the lateral direction, and θ[rad] represents yaw angle of the vehicle model.

2.2 Linearized Model

We consider the form of QP to solve this problem, so this nonlinear and non-convex model needs to be linearized.

The first-order Taylor expansion:

$$\begin{aligned}
\dot{x}(t) &= -\theta * v_0 * sin(\theta_0) + v * cos(\theta_0) + v_0 * \theta_0 * sin(\theta_0) \\
\dot{y}(t) &= \theta * v_0 * cos(\theta_0) + v * sin(\theta_0) - v_0 * \theta_0 * cos(\theta_0) \\
\dot{\theta}(t) &= \frac{1}{L} * (v * tan(\delta_0) + \delta * \frac{v_0}{cos^2(\delta_0)} - \delta_0 * \frac{v_0}{cos^2(\delta_0)})
\end{aligned} \tag{2.2}$$

$(v_0,\, \theta_0,\, \delta_0)$ is the operation point.

2.3 Discretized Model

Then we need to discretize the model for MPC. We set the time stamp $\triangle t$ as a constant parameter.

$$\begin{aligned}
x(k+1) &= x(k) + \dot{x}(k) * \triangle t \\
y(k+1) &= y(k) + \dot{y}(k) * \triangle t \\
\theta(k+1) &= \theta(k) + \dot{\theta}(k) * \triangle t
\end{aligned} \tag{2.3}$$

Therefore, the model can be represented:

$$\xi(k+1) = f_k(\xi(k), u(k)) \tag{2.4}$$

With the state vector $\xi = [x, y, \theta]^T$ and $u = [v, \delta]^T$.

3 Pedestrian Tracking

The following chapter discusses the method of pedestrian tracking. Section 3-1 is used to highlight the overall structure and describe the detection part. Following, the tracking algorithm is emphasized in section 3-2. The experimental results are listed in section 3-3. The chapter ends with the concluding remarks provided in section 3-4.

3.1 Process of tracking people

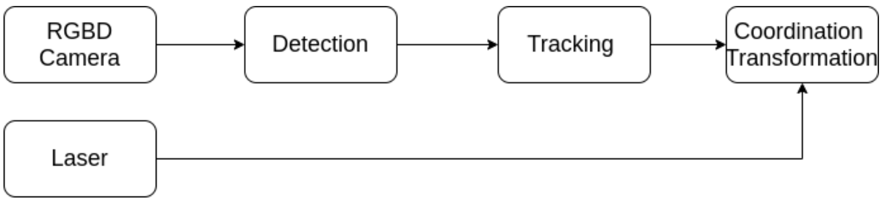


Figure 3.1: Process of tracking people

The first step is human detection that is done by deep CNN. The Yolo network is run on the GPU of Nidia Jetson Tx2.

Then, for the multi-people situations, the tracking algorithm is used to track and estimate speed of every people.

Lidar is used to compensate for the field of view limitation. At last, the 3D coordination is calculated by the depth provided by the RGBD camera and the projection matrix.

3.2 Tracking algorithm(SORT)

Simple online and real time tracking is a frame-to-frame tracking. It includes detection, estimation, and data association. It is the combination of the Kalman filter and Kuhn-Munkres algorithm. It guarantees speed and accuracy.

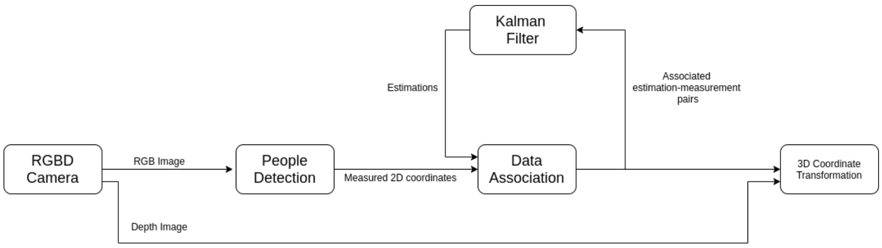


Figure 3.2: SORT structure

3.2.1 Kalman Filter

Here we describe the object model, i.e. the representation and the motion model used to propagate a target's identity into the next frame. We approximate the inter-frame displacements of each object with a linear constant velocity model which is independent of other objects and camera motion.

The state of each target is modelled as: $x = \begin{bmatrix} u & v & s & r & \dot{u} & \dot{v} & \dot{s} \end{bmatrix}^T$,

where u and v represent the horizontal and vertical pixel location of the centre of the target, while the scale s and r represent the scale (area) and the aspect ratio of the target's bounding box respectively. Note that the aspect ratio is considered to be constant.

The equation of states is Equation 3.1.

$$x' = Fx + \nu \tag{3.1}$$

$$\text{where } F = \begin{bmatrix} 1 & 0 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \nu = \begin{bmatrix} \frac{\triangle t^2}{2} & 0 \\ 0 & \frac{\triangle t^2}{2} \\ \frac{\triangle t^2}{2} & \frac{\triangle t^2}{2} \\ 0 & 0 \\ \triangle t & 0 \\ 0 & \triangle t \\ \triangle t & \triangle t \end{bmatrix} \begin{bmatrix} a_x & a_y \end{bmatrix}$$

The default value of acceleration terms a_x, a_y is 1 $\frac{pixel}{s}$.

Therefore the prior estimate covariance P and the covariance of the process noise Q could be calculated by Equation 3.2.

$$\begin{aligned} P' &= FPF^T + Q \\ Q &= \nu\nu^T \end{aligned} \tag{3.2}$$

The initial value of P is $P_{init} = diag(10, 10, 10, 10, 10000, 10000, 10000)$, since the acceleration terms and velocity terms are uncertain.

The second part of the Kalman filter is the update.

$$\begin{aligned} y &= z - Hx' \\ S &= HP'H^T + R \\ K &= P'H^TS^{-1} \\ x &= x' + Ky \\ P &= (I - KH)P' \end{aligned} \tag{3.3}$$

In the equation 3.3, z represents measurements(u,v,s,r). H represents the observation model. $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

R represents the covariance of the measurement noise. $R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$

3.2.2 Data association

We use Kuhn-Munkres algorithm to assign measurements and predictions, based on the maximal intersection over union(maximum IOU).

IOU \ Measurement / Prediction	M_1	M_2	M_3
P_1	0.4	0	0.8
P_2	0.3	0.2	0.4
P_3	0.16	1	0.9
P_4	1	0.1	0.7

Table 3.1: example of a assignment problem

In Table 3.1, each prediction-measurement pair has one value representing IOU. We need to choose a few pairs to get maximal IOU and each prediction and measurement has only to be chosen at most one time. Additionally, a minimum IOU is imposed to reject assignments where the detection to target overlap is less than IOU_{min}.

3.2.3 Creation and Deletion of Track Identities

When objects enter and leave the image, unique identities need to be created or destroyed accordingly. For creating trackers, we consider any detection with an overlap less than IOU_{min} to signify the existence of an untracked object. The tracker is initialised using the geometry of the bounding box with the velocity set to zero. Since the velocity is unobserved at this point the covariance of the velocity component is initialised with large values, reflecting this uncertainty. Additionally, the new tracker then undergoes a probationary period where the target needs to be associated with detections to accumulate enough evidence in order to prevent tracking of false positives.

Tracks are terminated if they are not detected for T_{Lost} frames. This prevents an unbounded growth in the number of trackers and localisation errors caused by predictions over long durations without corrections from the detector. In all experiments T_{Lost} is set to 1 for two reasons. Firstly, the constant velocity model is a poor predictor of the true dynamics and secondly we are primarily concerned with frame-to-frame tracking where object re-identification is beyond the scope of this work. Additionally, early deletion of lost targets aids efficiency. Should an object reappear, tracking will implicitly resume under a new identity.

3.3 Laser

When the mobile robot moves, it is probably that people are out of the field of view of the camera. This would cause unstable and inconsistent behaviour of robot. Therefore, Laser is used as a complementary of the camera. The laser we use is one-layer and could cover 270 degree. The laser could be used to memorize positions of people.

The preprocess of data from laser we do is segmentation. The figure 3.3 shows the result of segmentation.

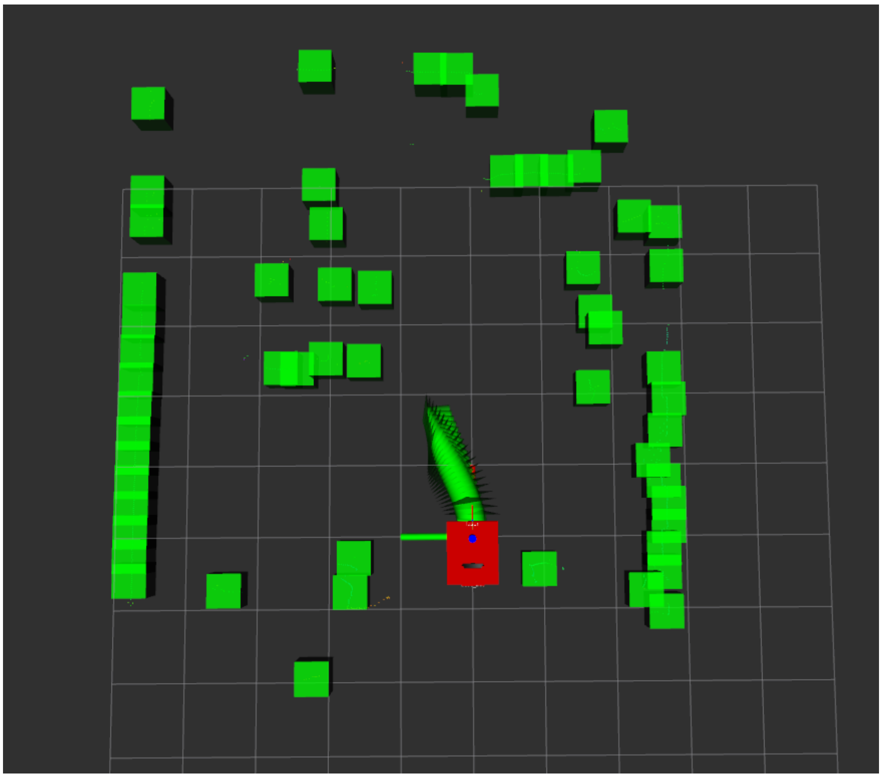


Figure 3.3: Segmentation of laser data

3.4 Experimental results

The tracking algorithm is tested in simulation environment and the real environment. Figure 3.4 shows the test in the software Gazebo. Everyone is labelled by a different number.

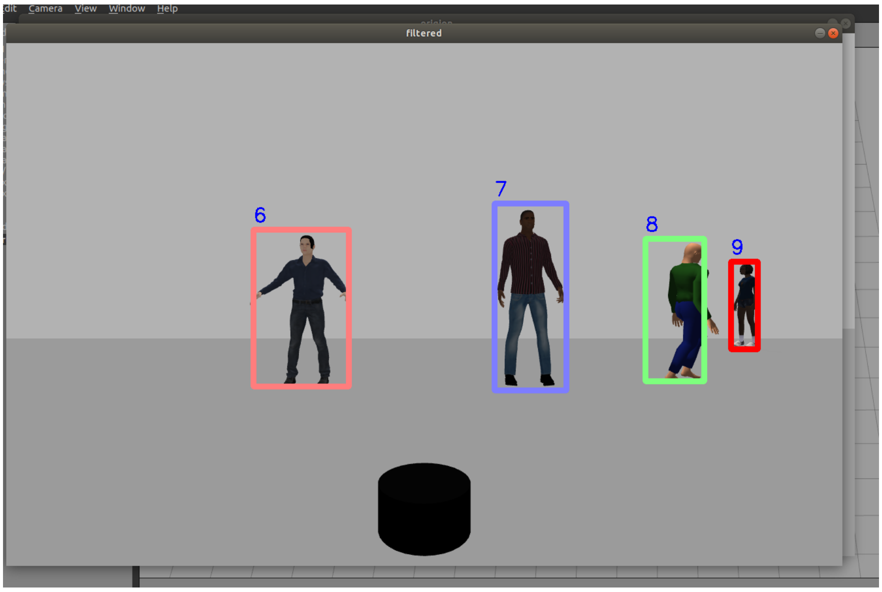


Figure 3.4: tracking in simulation environment

Figure 3.5 uses Realsense D435 camera to get pictures. Everyone is assigned a unique number, and this number would be kept unless the person is out of the field of view.

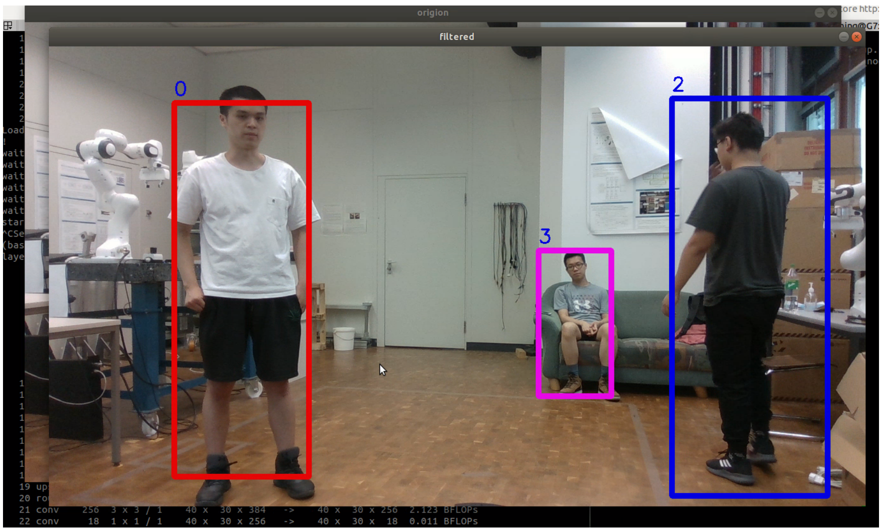


Figure 3.5: tracking in real environment

4 Collision avoidance of the mobile robot

In this chapter the methods for the controller for trajectory generation is discussed. The control loop of motion planning is introduced in section 4-1. control objectives and practical constraints for the controller are defined in section 4-2, and 4-3 respectively. The optimal control problem is formulated in section 4-4 to form the core of the optimization based controller. Consequently,the important part of the trajectory generation operation is elaborated with the collision avoidance constraints. The approaches to formulate the collision avoidance constraints offer insight into the working of the MPC based trajectory generator. Section 4-5 further explains the optimization algorithm and sequential convex method for the application.

4.1 Motion planning

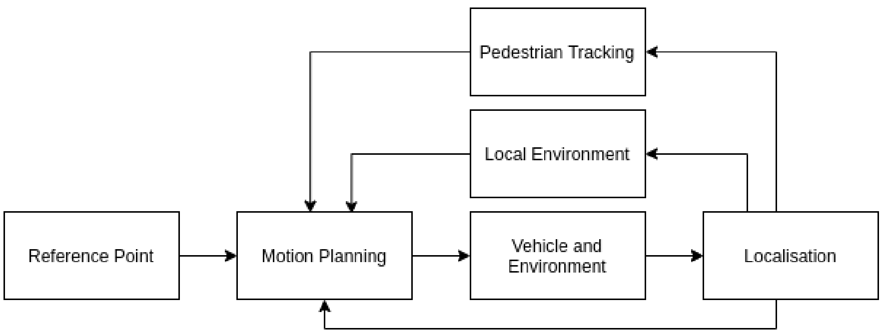


Figure 4.1: Control loop

The control structure is in Figure 4.1. The motion planning part is the topic in this section. The MPC model would be introduced to drive the vehicle to the reference point, with the information of the vehicle and people and local environment.

4.2 Control Objectives

Trajectory generation with respect to automotive systems primarily deals with the safe trajectory generation while satisfying the vehicle and environment constraints. We also pay more attention to the pedestrian. We don't want the behavior of the AGV would influence surrounding people. The previously described system architecture allows us to formulate the requirements of the controller. The following control objectives were defined to form a framework for the controller development. The controller needs to be able to,

- Generate safe trajectory reference in the motion planner level.
- Maintain the vehicle velocity, try to avoid stop situations.
- Involve trajectory prediction of human to avoid unreasonable trajectories.

4.2.1 Practical constraints

Position
The position of the vehicle must not collide with obstacles, including pedestrians.

Velocity
The speed objective is bounded by the 0.3[m/s], because the AGV is supposed to be working in the indoor environment.

Steering angle
The steering angle has a limitation in reality. We choose the range from -40 degree to 40 degree, in order to get a more smooth path.

Acceleration
Acceleration limits mainly satisfy the consideration on passenger comfort and vehicle actuator limits in real life. Acceleration metrics is considered in the objective function.

Non-holonomic constraints
The equation of the linearized kinematic model 2.4 is taken as the constraint, so that the trajectory satisfy the non-holonomic vehicle constraint.

4.3 Collision avoidance constraints

The formulation of the collision avoidance constraints is a crucial step to generate collision free trajectories. The performance index of the optimal control problem is formulated to achieve the best desired behaviour based on the objectives. The optimal control problem formulation to achieve the best performance with the control objectives and the following practical constraints respect the physical and design constraints of the vehicle. The additional constraints required to achieve safe collision free trajectories are added to the optimization problem formulation.

In this section, the focus is placed on formulating the feasible region for the controlled vehicle to travel in the driving scenario. This creates a non convex optimization problem with respect to the safe search space for trajectory generation.

4.3.1 Representation of collision avoidance

At the time $t + k$, the position of the vehicle is not in the region of obstacles.

$$z(t+k) \notin \mathcal{O}(t+k), \tag{4.1}$$

where $z(t + k)$ means the position of the vehicle at t+k and $\mathcal{O}(t + k)$ represents the area of obstacles at $t + k$.

The equation 4.1 could be formulated

$$\|z(t+k) - p_o(t+k)\| \geq d_{safe}, \tag{4.2}$$

where $p_o(t + k)$ means the position of the obstacle at time t+k.

It is the non convex constraint and thus it is NP-hard to solve. All known algorithms to solve a non-convex QCQP have a time complexity that grows exponentially with the dimensions of the optimization problem. Therefore, the sequential quadratic programming method is involved.

4.4 Optimal control problem formulation for trajectory generation

The MPC is defined as a control center which has all information about the vehicles and obstacles, their locations, and their reference trajectories. The control center should solve an optimal control problem by considering the benefits of all vehicles and constraints applied on them. The constraints include the vehicle dynamic model, vehicle collision avoidance between each vehicle and obstacle, and constraints on the control input. The solution is system-wide optimal, since the whole optimization problem is bulkly solved.

The control objectives and practical constraints defined give an guideline for the objective function and design constraints. The collision avoidance terms, define the optimization search space for planning safe collision free trajectories. The following equation is the OCP problem formulation,

$$\begin{aligned} min\ z^T(k+N)Pz(k+N) + \sum_{i=0}^{N-1} z^T(k+i)Qz(k+i) + u^T(k+i)Ru(k+i) \\ +\triangle u^T(k+i)W\triangle u(k+i) \end{aligned} \tag{4.3}$$

subject to

$$\xi(k+i) = f_k(\xi(k+i-1), u(k+i-1)) \tag{4.4}$$

$$z(k+i) = C\xi(k+i) \tag{4.5}$$

$$z(k+i) \notin \mathcal{O}(k+i), \tag{4.6}$$

$$u(k+i) \in \mathcal{U} \tag{4.7}$$

$$z(k+i) \in \mathcal{Z} \tag{4.8}$$

Equation 4.3 is the objective function that minimizes the distance between the vehicle position and the reference trajectory over the prediction horizon. The objective function also minimizes the control input variations and input. Equations(4.4-4.8) show the constraints. Equation 4.4 indicates the dynamic model of the vehicles. In 4.5, z_i represents the system output, here it is the vehicle's position and orientation. Equation 4.7 specifies the control input boundary condition. Equation 4.8 specifies the state space boundary condition. Equation 4.6 indicates the obstacle collision avoidance.

The following section is designed to simplify the optimization problem to a quadratic constrained quadratic programming (QCQP) framework.

4.5 Sequential convex programming

The main idea of SCP is to convexify the non-convex parts of the objective function and the constraints, and preserve their convex parts. The principle procedure is to solve a sequence of convex optimization problems.

4.5.1 Convex Approximations for static obstacles

The non-convex constraint in equation 4.2,

$$(x(k) - x_{obs}(k))^2 + (y(k) - y_{obs}(k))^2 \geq d_{safe}^2$$

Algorithm 1 SCP algorithm to solve the non-convex collision avoidance optimization problem

```
1: c := 1   { Iteration counter }
2: Determine starting point e_c
3: Compute the objective value of the non-convex program J_c using e_c
4: Form a convex approximation of the non-convex parts of the inequality constraints using e_c
5: Compute the optimal solution e_{c+1} of the resulting convex program
6: Compute the objective value of the non-convex program J_{c+1} using e_{c+1}
7: if J_c − J_{c+1} ≤ ϵ then
8:     return e_{c+1};
9: end if
10: c := c + 1
11: goto 4
```

The convex approximations using the first order Taylor-expansion

$$(x_{ite}-x_{obs})(x-x_{obs})+(y_{ite}-y_{obs})(y-y_{obs}) \geq \frac{(d_{safe}^2 + (x_{ite} - x_{obs})^2 + (y_{ite} - y_{obs})^2)}{2}, \tag{4.9}$$

where (x_{ite}, y_{ite}) is the operation point calculated in the previous step.

Geometric Interpretation of constraint linearization is that the constraint boundary is always orthogonal to the line between the vehicle and the obstacle.

SCP is a restriction of the non-convex QCQP. The convex program can be infeasible.

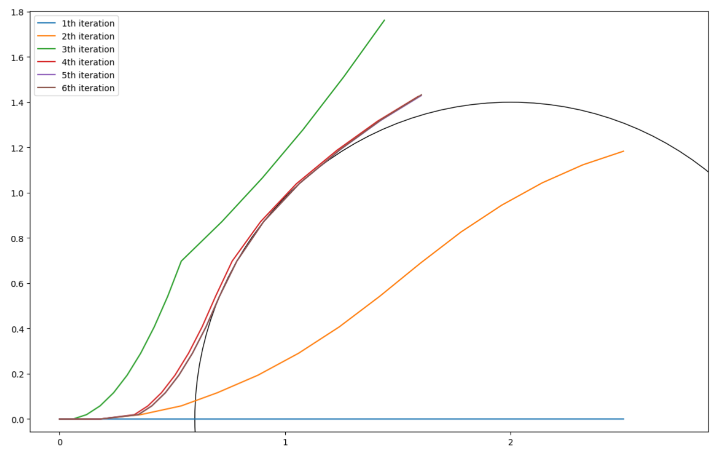


Figure 4.2: iterations of SCP

4.5.2 Convex Approximations for people

In order not to influence people, we have to take the possible trajectories of people into account.

The non-convex constraint of moving or static people is
$\{x(k+i)-x_{obs}(k+i)-v_{obs}^{x}\triangle t\times i\}^2+\{y(k+i)-y_{obs}(k+i)-v_{obs}^{y}\triangle t\times i\}^2 >= d_{safe}^2$

The convex approximations using the first order Taylor-expansion is

$$(x_{ite}-x_{obs}-v_{obs}^{x}t_i)(x-x_{obs}-v_{obs}^{x}t_i)+(y_{ite}-y_{obs}-v_{obs}^{y}t_i)(y-y_{obs}-v_{obs}^{y}t_i)\geq \{d_{safe}^2+(x_{ite}-x_{obs}-v_{obs}^{x}t_i)^2+(y_{ite}-y_{obs}-v_{obs}^{y}t_i)^2\}/2, \tag{4.10}$$

where $t_i = \triangle t \times i$.

We predict the position of people at each step based on estimated velocity, and form collision avoidance constraints.

4.5.3 Feasibility of the Approximate Program

Depending on the starting point, the approximate convex program can be infeasible. To overcome this issue, a slack variable ρ could be added in the linearsized constraints4.9 and 4.10 and the cost function 4.3. Then, the program is always feasible and ρ would be minimized through iterations.

The slack variable ρ corresponds to the maximum violation of the inequality constraints and therefore, this is equivalent to the exact penalty method with coefficient of ρ large enough. This means that constraint violations will not occur unless there is no feasible solution to the original program.

5 Experiments and discussions

5.1 Experiments in the simulation environment

We use Gezebo to build the simulation environment. The mobile robot and pedestrian are modelled.

Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. This is a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces.

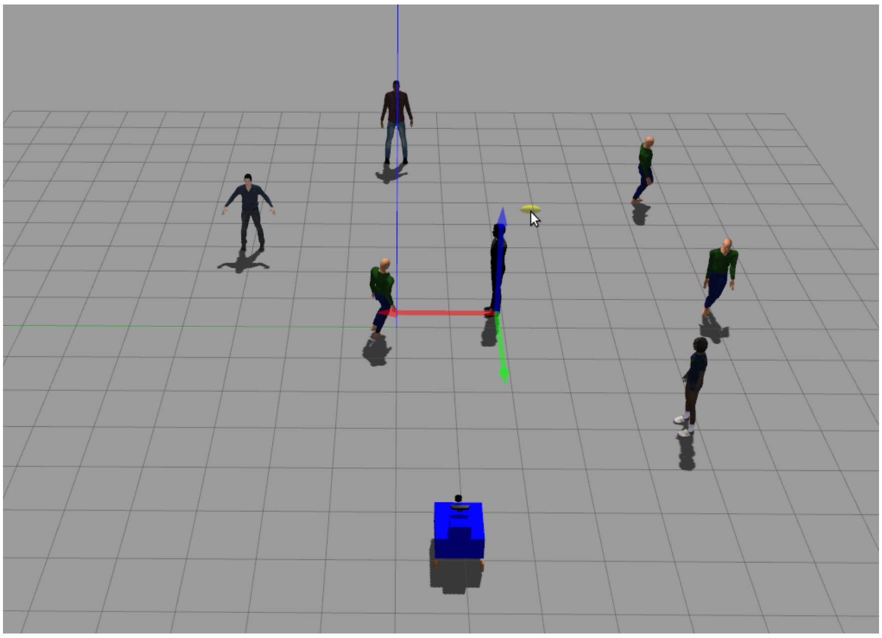


Figure 5.1: Simulation environment

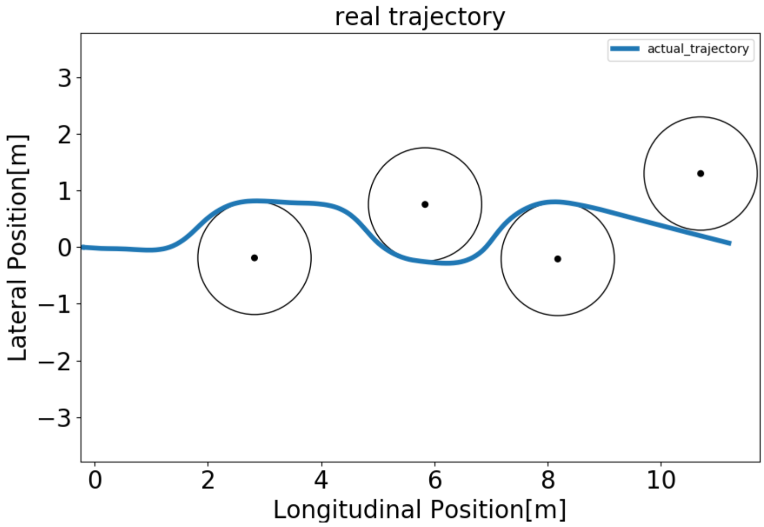


Figure 5.2: Trajectory in simulation

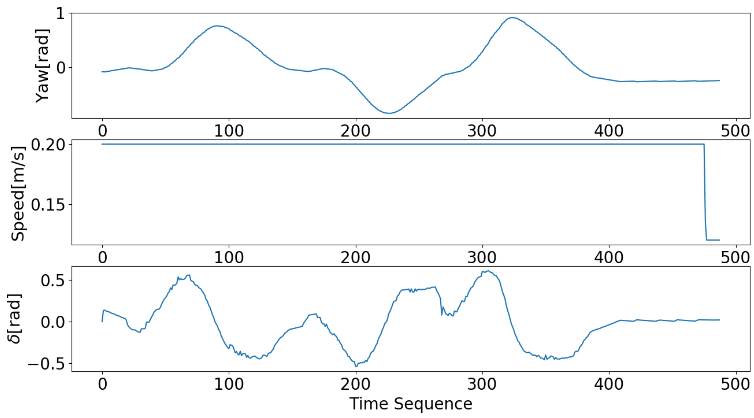


Figure 5.3: Control input in simulation

In figure 5.2, the dot represents the person. The circle around the dot depicts the forbidden area for the mobile robot. It is clear that the trajectory generated is smooth and reasonable. In figure 5.3, the control input speed shows MPC makes the decision to move to the desired position as soon as possible.

5.2 Experiments in the real environment

The mobile robot we use is in figure 5.4.

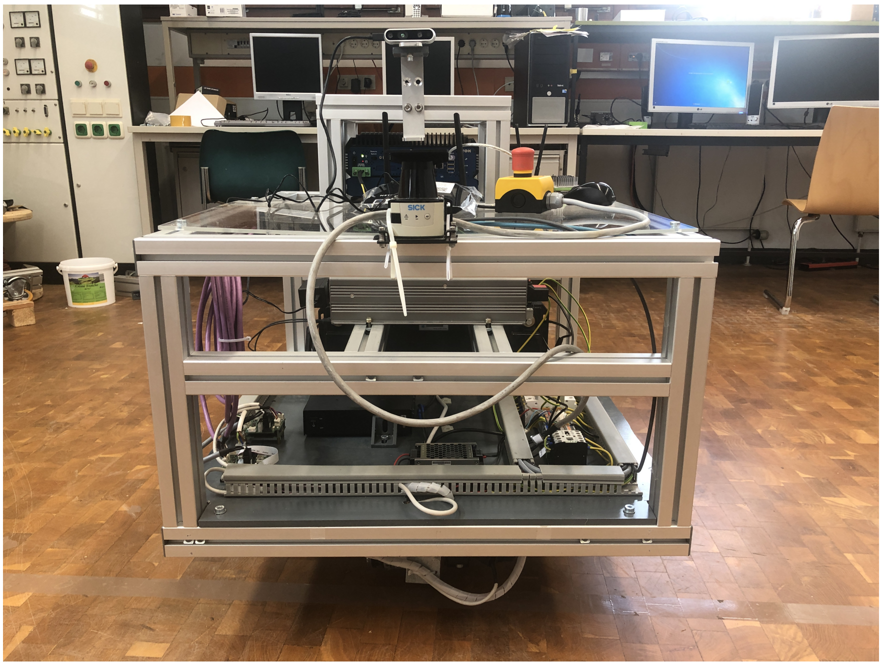
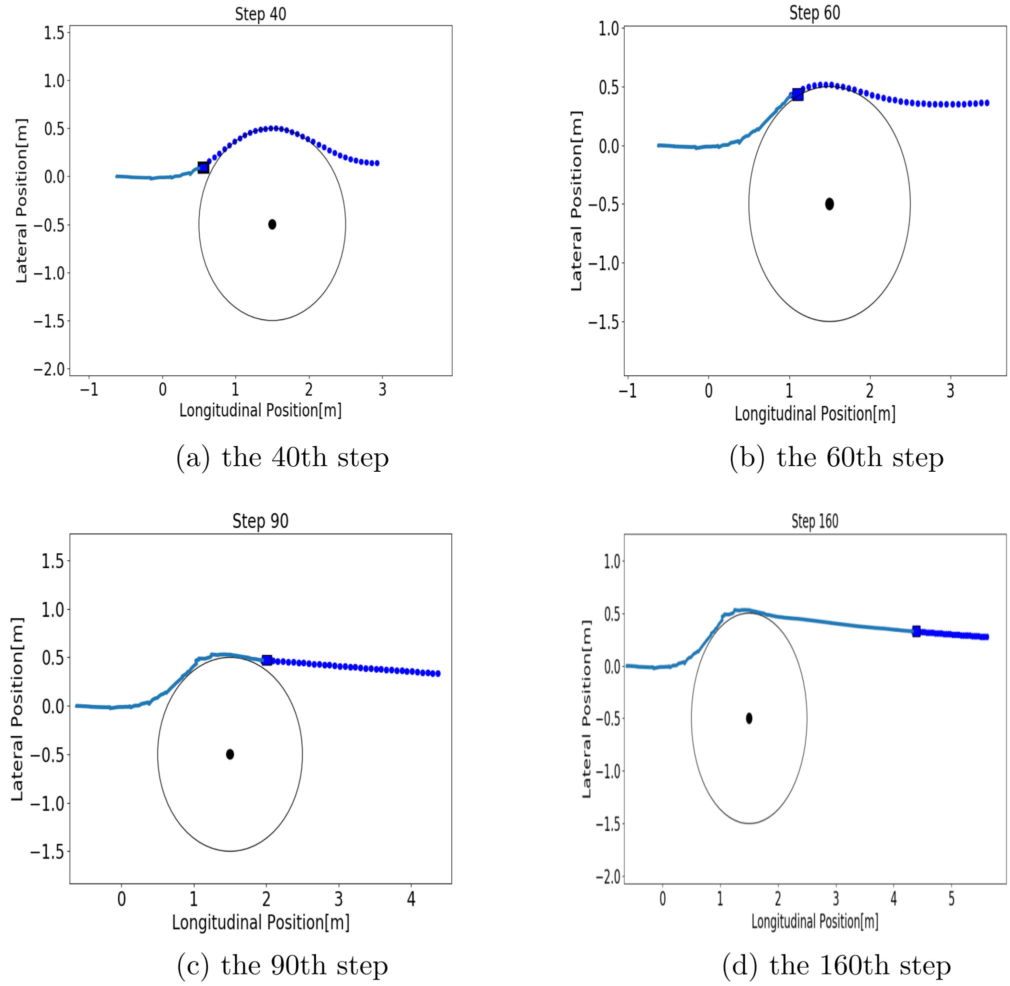


Figure 5.4: The mobile robot

5.2.1 Experiment of static obstacles

In the experiment, we set the virtual obstacles and control the real robot.



(a) the 40th step

(b) the 60th step

(c) the 90th step

(d) the 160th step

Figure 5.5: Steps of the trajectory

In figure 5.5, There are four steps of trajectories. The dotted lines are reference trajectories by MPC and the solid lines are actual trajectories. We can see the mobile robot could track the desired trajectory very well based on one-layer MPC in the slow-speed condition.

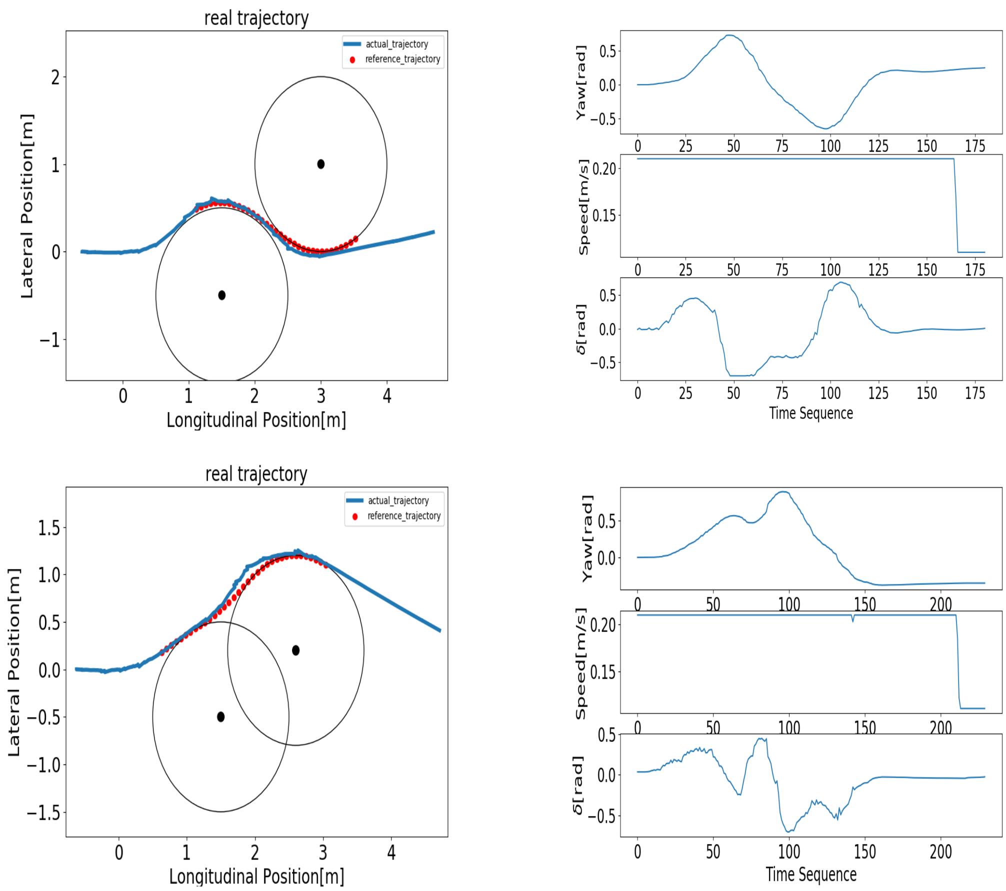
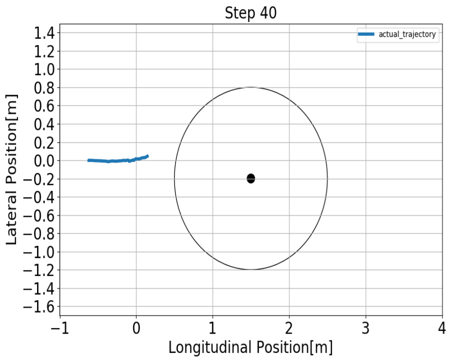


Figure 5.6: real and reference trajectories and control effort

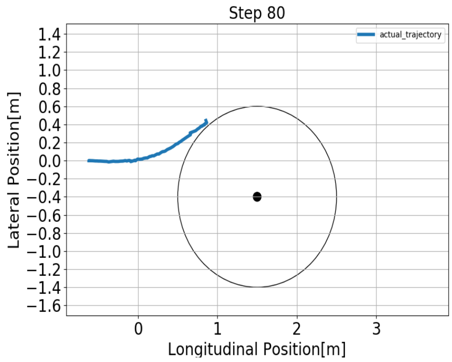
We also tested two situations in figure 5.6 in which two people stand ahead the mobile robot, and the reasonable reference trajectory represented by a red dotted line is generated which does not violate constraints and the actual trajectory represented by blue line could follow the reference one. The disadvantage in this experiment is that when the mobile robot has a big yaw angle, it would has deviate between the actual trajectory and reference one, due to the linearisation. The linearized kinematic model brings error.

5.2.2 Experiment of dynamic obstacles

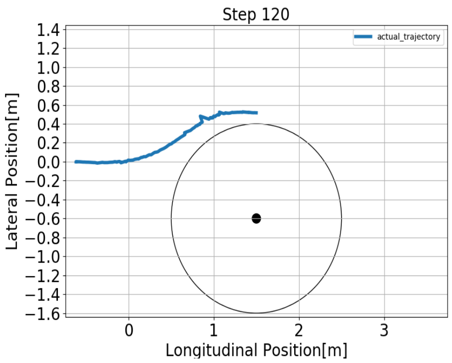
We tested artificial moving obstacles that go down, up, left.



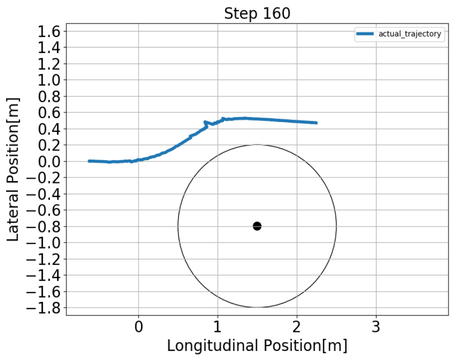
(a) the 40th step



(b) the 80th step



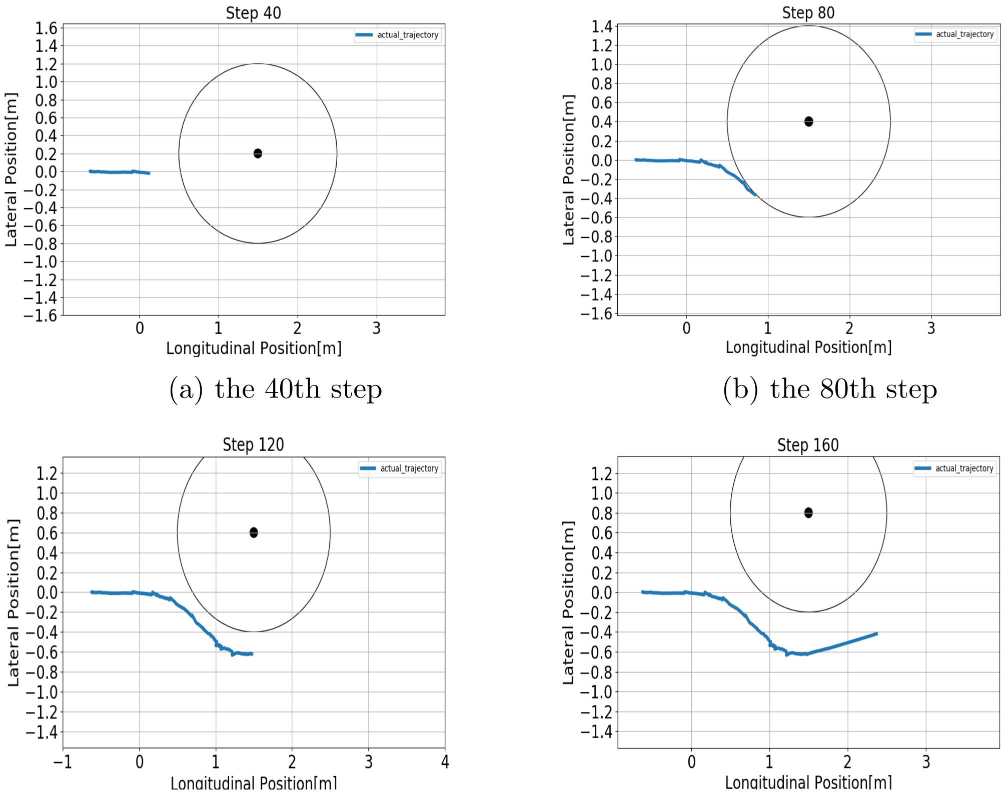
(c) the 120th step



(d) the 160th step

Figure 5.7: Steps of the trajectory to avoid the obstacle moving down

In figure 5.7, the obstacle moves down and MPC gives the trajectory that goes up.



(a) the 40th step

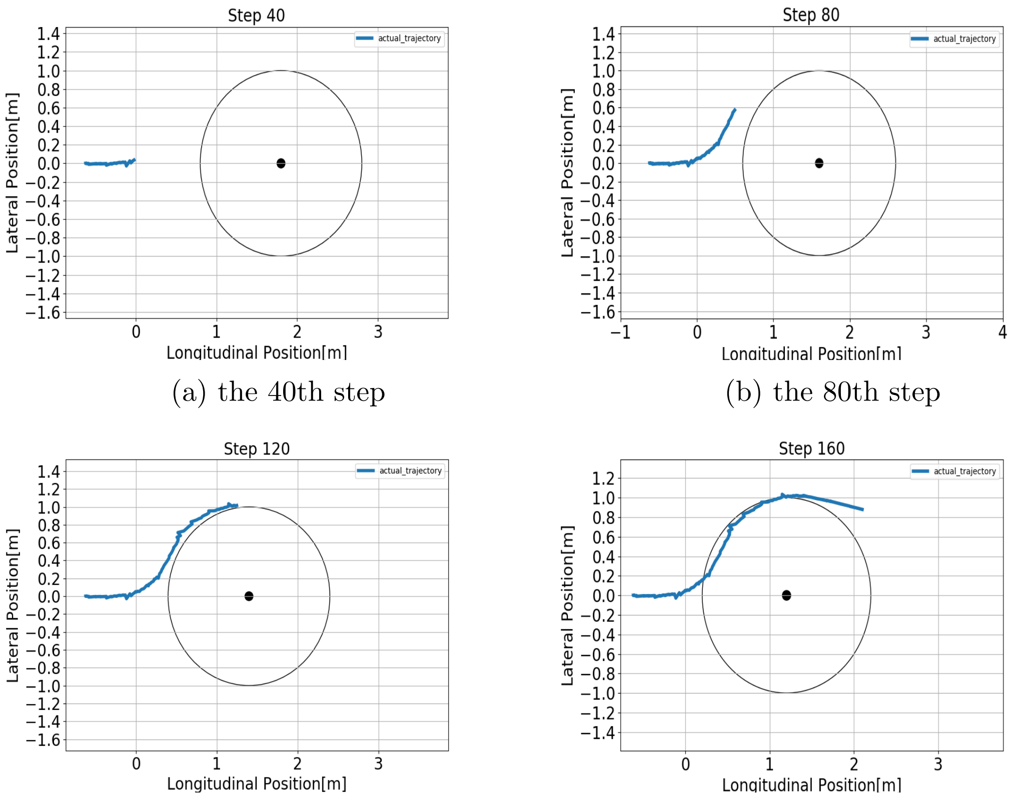
(b) the 80th step

(c) the 120th step

(d) the 160th step

Figure 5.8: Steps of the trajectory to avoid the obstacle moving up

In figure 5.8, the obstacle moves up and MPC gives the trajectory that goes down.



(a) the 40th step (b) the 80th step

(c) the 120th step (d) the 160th step

Figure 5.9: Steps of the trajectory to avoid the obstacle moving towards the mobile robot

In figure 5.9, the obstacle moves towards the mobile robot and MPC gives the trajectory to avoid collision early. In the 5.9b, the mobile robot has turned.

6 Conclusion and future work

6.1 Conclusion

We implemented perception, motion planning, control algorithm for the mobile robot.

- The mobile robot could distinguish different people and estimate speed
- The mobile robot could avoid them and does not influence them, due to consideration of velocity of people.

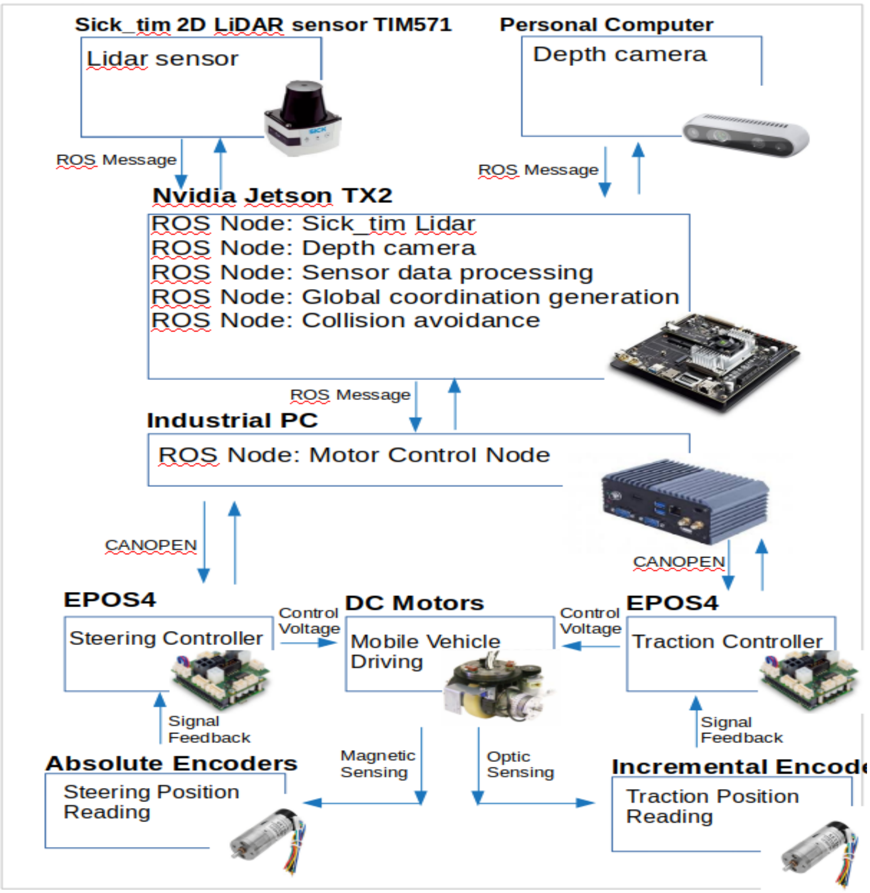


Figure 6.1: Framework

Then whole framework is in figure6.1. We are glad to have this accomplishment and have a chance to practice theories and what we learned.

6.2 Future work

There are a few jobs that could do to improve performance.

- The tracking algorithm could use model-free method, like LSTM RNN.
- The compensation for the perception due to the robot motion should be considered.
- The NMPC could be used, to have more a accurate model instead of linearisation method.

Bibliography

[AMA14] Alrifaee, Bassam ; Mamaghani, Masoumeh G. ; Abel, Dirk: Centralized non-convex model predictive control for cooperative collision avoidance of networked vehicles. In: *2014 IEEE international symposium on intelligent control (ISIC)* IEEE, 2014, S. 1583–1588

[BGO+16] Bewley, Alex ; Ge, Zongyuan ; Ott, Lionel ; Ramos, Fabio ; Upcroft, Ben: Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)* IEEE, 2016, S. 3464–3468

[KB+91] Koren, Yoram ; Borenstein, Johann u. a.: Potential field methods and their inherent limitations for mobile robot navigation. In: *ICRA* Bd. 2, 1991, S. 1398–1404

[QK93] Quinlan, Sean ; Khatib, Oussama: Elastic bands: Connecting path planning and control. In: *[1993] Proceedings IEEE International Conference on Robotics and Automation* IEEE, 1993, S. 802–807

[RF18] Redmon, Joseph ; Farhadi, Ali: Yolov3: An incremental improvement. In: *arXiv preprint arXiv:1804.02767* (2018)

[SNS11] Siegwart, Roland ; Nourbakhsh, Illah R. ; Scaramuzza, Davide: *Introduction to autonomous mobile robots.* MIT press, 2011

[SPAD19] Schoels, Tobias ; Palmieri, Luigi ; Arras, Kai O. ; Diehl, Moritz: An NMPC Approach using Convex Inner Approximations for Online Motion Planning with Guaranteed Collision Avoidance. In: *arXiv* (2019), S. arXiv–1909