Semester Project

Algorithm to Pass the First Lap for Autonomous Student Formula

Dmytro Khursenko

Supervisor: Ing. Jan Čech, Ph.D.

Faculty of Electrical Engineering

Department of Cybernetics

January 20, 2023

Table of Contents

Chapter 1
Introduction

This semester project has a focus on an algorithm that allows a driverless formula to pass the first lap of the Formula Student Driverless competition. The algorithm includes a reactive path planning algorithm and a speed profile algorithm for a driverless vehicle in the first lap. The aim is to develop a speed profile algorithm that has better results than constant speed profile with velocity 5 m/s. To prove the results of a developed algorithm, the speed profile will be compared with the constant speed profile and with the optimal speed profile that knows the whole track.

Formula Student is an international student engineering competition. The competition is designed to develop and showcase the engineering skills of students by challenging them to design, build and race a single-seat race car. In 2017, Formula Student Germany introduced a Driverless class, where the teams compete with their autonomous vehicles. One of the difficulties in the driverless competition is that the vehicle does not know the whole track before the start. Thus the autonomous vehicle should make decisions in an unknown environment on the first lap.

eForce FEE Prague Formula is a Formula Student team from the Czech Republic. The team is under the Faculty of Electrical Engineering of the Czech Technical University in Prague. eForce FEE Prague Formula is the first Czech team in the Formula Student competition that successfully build a formula with electric drive and autonomous control. In 2020 the team introduced the first driverless formula in the Czech Republic – DV.01. I joined the team in 2021 and I have been working on a path planning. After the 2022 race season, the team figured out that our driverless formula is not so fast and set a goal to work on a speed profile. The project will introduce a speed profile algorithm for the first lap.

Chapter 2
Autonomous System

An autonomous system is the main feature of DV.01. It is responsible for everything from observing the environment using sensors, planning a trajectory and sending the right commands to the car's actuators to correctly follow a trajectory. The data input is provided by the sensors: Ouster OS1 LiDAR, a Stereolabs ZED camera, and an SBG Systems Ellipse2-D inertial navigation system (INS). In Figure 2.1 you can see the driverless formula DV.01 with the sensors.

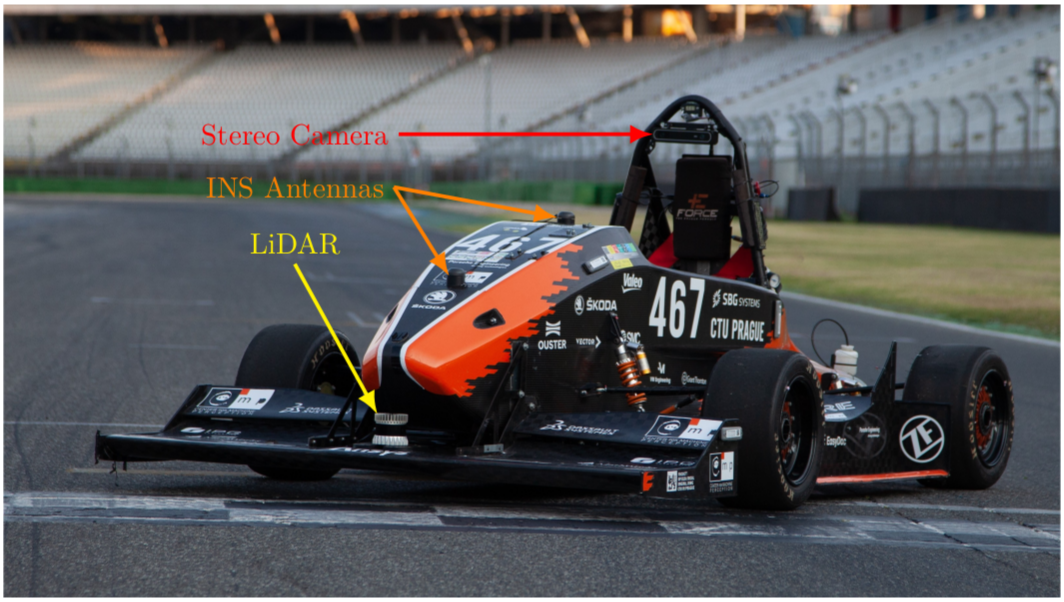

Figure 2.1: Photo of the autonomous formula DV.01 with the sensors.

Section 2.1
Autonomous Pipeline

Besides the sensors, the autonomous system includes an autonomous pipeline that takes a big role in driverless abilities of DV.01. It runs on a Zotac Z-Box computer equipped with a standard Linux Ubuntu OS. The autonomous pipeline consists of separate modules. For communication between the modules we implemented our own system that is based on Robot Operating System (ROS). A diagram of the modules and their connections is shown in Figure 2.2.

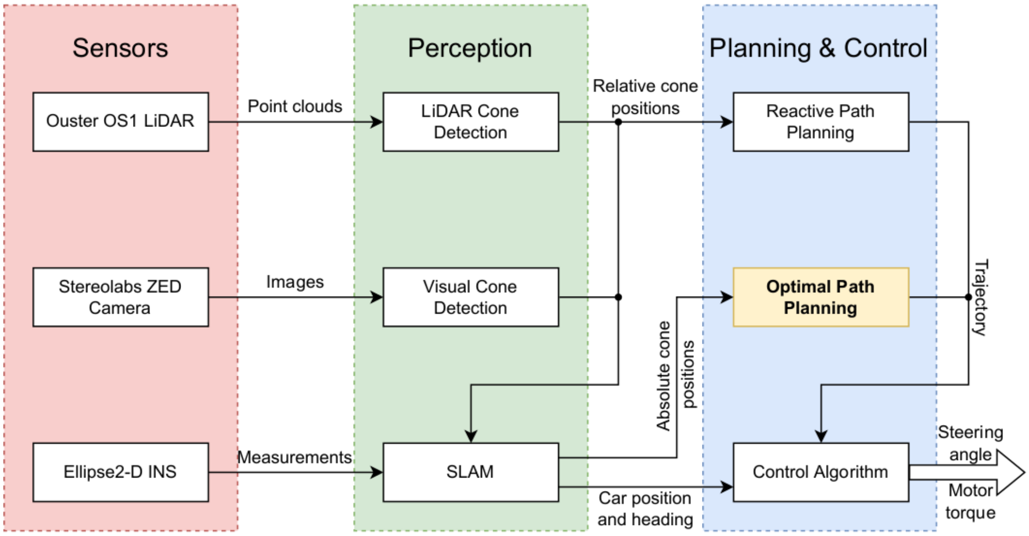

Figure 2.2: A diagram of the autonomous pipeline.

The autonomous pipeline starts with the data input provided by the sensors. RGB images from camera are processed by convolutional neural network YOLOv3 [1]. The neural network detects the cones in an image. After the cone detection, the cone pixels are subsequently projected into the world coordinates via a homography mapping between the image and the world [2]. Another option how to detect the cones is using LiDAR detector [3], also we are using the LiDAR for camera calibration. The inertial navigation system (INS) provides the position, orientation and speed of the formula. The measurements from INS and the relative cone positions from camera or LiDAR go to simultaneous localization and mapping algorithm (SLAM) [4]. The output of the SLAM is the absolute cone positions that are used to generate optimal path planning [5]. For the first lap the reactive path planning is used. It generates a center line trajectory between the blue and yellow cones. The end of the autonomous pipeline is Stanley control algorithm [6], where a steering angle and motor torque are set based on the generated trajectory, heading and position of the car. Finally, the steering angle and motor torque commands are sent via controller area network (CAN) bus to the appropriate electronic control units that execute the commands.

Chapter 3

Algorithm for the First Lap

There are different dynamic disciplines in the Formula Student competition. As an example the autocross is the dynamic discipline where a driverless formula has to finish only one lap. While in the trackdrive mission, the driverless formula has to finish 10 laps. Therefore, we use different driving strategies for every mission. For the first lap of trackdrive we use reactive path planning and after the first lap is completed, the optimal trajectory and optimal speed profile [5] are generated for the next nine laps.

In autocross there is an unknown environment for the car, so we need to have a robust algorithms to successfully finish this discipline. The reactive path planning is used to generate a conservative path. But to win the race the formula needs to drive fast, so there is a need in a speed profile.

Section 3.1

Reactive Path Planning

During the autocross and the first lap of trackdrive, the car relies on a reactive path planning algorithm. We have developed Algorithm 1 to find the center line between the blue and yellow cones. Let B be the set of blue cones and Y is the set of yellow cones that we detected. Also, we remember the blue and yellow cones that were used to find a center. Let $\vec{s}_0 = (x_0, y_0)$ be the start coordinate of the car. We obtain the final center line by interpolation of the elements of $\{\vec{s}_i\}_{i=1}^{N}$.

Algorithm 1 Reactive path planning algorithm

$i = 1$
while path not connected **do**
  Use the last used blue and yellow cones to find the line parameters (k,c).
  Separate the cones that are in front of the car

$$\hat{B} = \{\vec{b} = (x, y) \in B : sign(y - k \cdot x - c) = 1\}$$
$$\hat{Y} = \{\vec{y} = (x, y) \in Y : sign(y - k \cdot x - c) = 1\}$$

  Find the closest blue and yellow cone

$$\hat{\vec{b}} =_{\vec{b} \in \hat{B}} \|\vec{s}_i - \vec{b}\|$$
$$\hat{\vec{y}} =_{\vec{y} \in \hat{Y}} \|\vec{s}_i - \vec{y}\|$$

  Calculate center $\vec{s}_{i+1}$ between $\hat{\vec{b}}$ and $\hat{\vec{y}}$
  $i = i + 1$
end while

Section 3.2
Speed Profile

Motion is a complex process and nonlinear physical laws are applied in the real world that are hard to model. For the purpose of this project some physical laws and models are neglected: the tire model, the slip ratio and aerodynamic force. For now the model of the car is quite simple. It is a single tire with combined force transmission, acceleration capabilities of four tires and mass of the car. The friction ellipse 1 is used to describe the driving limits of the formula.

$$F_x^2 + F_y^2 \leq (\mu F_z)^2 \tag{1}$$

Where F_x is a longitudinal force, F_y is a lateral force, μ is a friction coefficient that describes the tire-road interaction and F_z is the normal force that presses the car to the road. The inequality 1 shows that the longitudinal and lateral forces have to remain within the handling limits defined by μF_z.

A speed or velocity profile is used to determine the most efficient way to complete a course in the shortest amount of time. In this chapter we will use the algorithm that is based on the “three-pass” approach described by Kapania [7]. For this project the implementation of the algorithm is taken from Horacek’s thesis [5]. Horacek implemented an optimal speed profile for the whole track that is perfectly known.

At the first lap the track is unknown for the car, we need to make some changes in the algorithm discussed earlier. The first change is the initial speed, because the speed is changing every frame. The next change is due to unknown environment – the algorithm should set a safe speed at the end of the local track. According to the rules [8] the minimum outside diameter of the turn is 9m, so the minimum radius is 4.5m. From Equation (2) we can calculate a safe speed.

$$v = \sqrt{\frac{\mu(m \cdot g + F_{aero}) \cdot r}{m}} \tag{2}$$

The formula’s parameters are taken from Table 1. The other variables for our case are: $r = 4.5m$, $g = 9.8\ m/s^2$ and $F_{aero} = 0$, because as we mentioned before - we neglect the downforce. Therefore, the rounded result is 5.75 m/s.

Parameter	Symbol	Value	Unit
Friction coefficient	μ	0.75	—
Formula mass	m	212	kg
Coefficient of lift	C_L	-3.82	—
Coefficient of drag	C_D	1.49	—
Reference aerodynamic area	A_{ref}	1.19	m^2
Maximum acceleration	a_{max}	2	m/s^2
Maximum deceleration	a_{min}	-4	m/s^2

Table 1: Physical parameters of DV.01 for the speed profile. Properties of the aerodynamic package come from an unpublished engineering design document [9].

When all parameters are known for the speed profile, we can introduce the algorithm. As mentioned before, the algorithm has three passes. The first pass or Algorithm 2 is calculating the maximum speed limited by the friction ellipse 1

Algorithm 2 First pass (maximum limited speed)

Input: Path $(\boldsymbol{p_i})_{i=0}^{P}$, path curvature $k(s)$ at the point $\boldsymbol{p_s}$
Output: Speed profile U'_x limited by a friction ellipse and safe speed limit
for $s = 0$ **to** P **do**
$U'_x(s) = \sqrt{\frac{\mu g}{k(s)}}$
end for
Set the safe speed 5.75 m/s to the last index of U'_x

The output of the first pass guarantees that the formula drives within the lateral and longitudinal limits. The second pass or Algorithm 3 adjusts the speed profile to acceleration limits.

Algorithm 3 Second pass (acceleration)

Input: Speed profile U_x, path $(\boldsymbol{p_i})_{i=0}^{P}$
Output: Speed profile U'_x limited by acceleration capabilities
for $s = 1$ **to** P **do**
$c = 2a_{max}||\boldsymbol{p_s} - \boldsymbol{p_{s-1}}||$
$U'_x[s] = \min\left(U_x[s], \sqrt{U_x[s-1]^2 + c}\right)$
end for

The third pass or Algorithm 4 is responsible for smooth breaking and as an input takes the speed profile after the second pass.

Algorithm 4 Third pass (braking)

Input: Speed profile U_x, path $(\boldsymbol{p_i})_{i=0}^{P}$
Output: Speed profile U'_x limited by braking capabilities
for $s = P$ **to** 1 **do**
$c = 2a_{min}||\boldsymbol{p_s} - \boldsymbol{p_{s-1}}||$
$U'_x[s-1] = \min\left(U_x[s-1], \sqrt{U_x[s]^2 + c}\right)$
end for

At the end of the third pass, we have a speed profile for one frame. The next question is what speed should be set. It will be a future work to find out the answer to this question, but for now we will take the maximum velocity of the speed profile. In the next chapter the experiments and results of the speed profile will be presented.

Chapter 4
Experiments

The important component of the experiments are the testing data. For the correct algorithm validation, the data were collected from the real world testing to have nearly the same conditions as at Formula Student Driverless competition.

Before the speed profile can be integrated with the formula's software, there is a need to test and validate the algorithm in a simulator. In Section 4.1 we will describe our simulator. After, the tests and results will be demonstrated in the next sections.

Section 4.1
Simulator

The simulator Virtual Milovice 4.1 has been created by eForce members at the end of 2022. It simulates the functionality of DV.01. The whole autonomous pipeline from Section 2 is implemented in the simulator. The simulator is quite new and some advanced physics laws are not implemented yet e.g. car slipping. But it has enough functionality to test the speed profile.

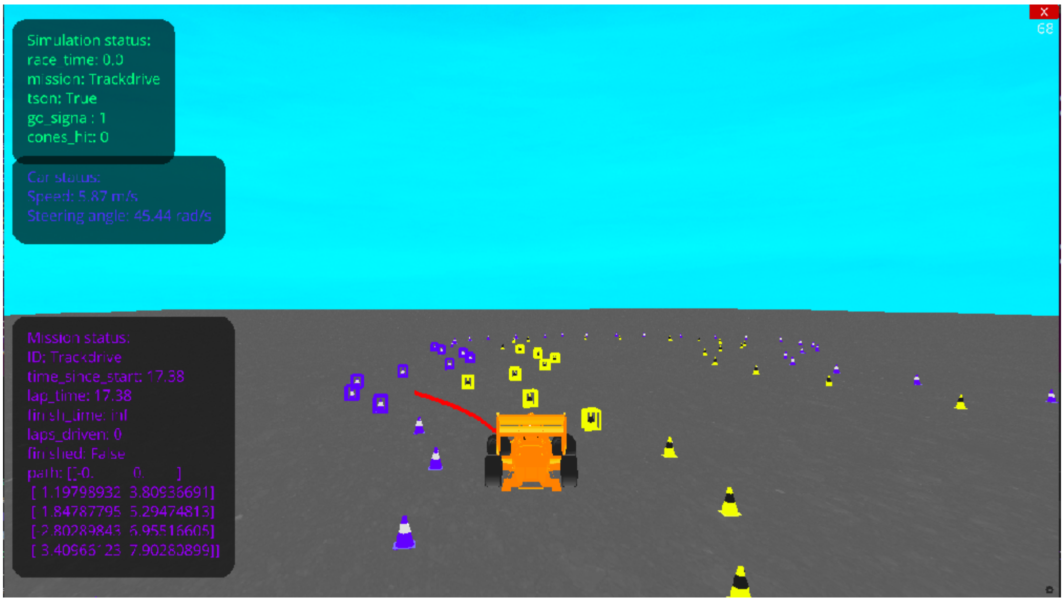

Figure 4.1: DV.01 in the simulator during the trackdrive mission.

Section 4.2

Local Track

The local track is defined as unknown environment or one camera frame. It is a situation where the formula observes the world in front of itself. Typically it is a 10-15m long track. For the local test we will have two tests: with a zero initial speed 4.2 and with non-zero initial speed 4.3.

Subsection 4.2.1

Local Track with a Zero Initial Speed

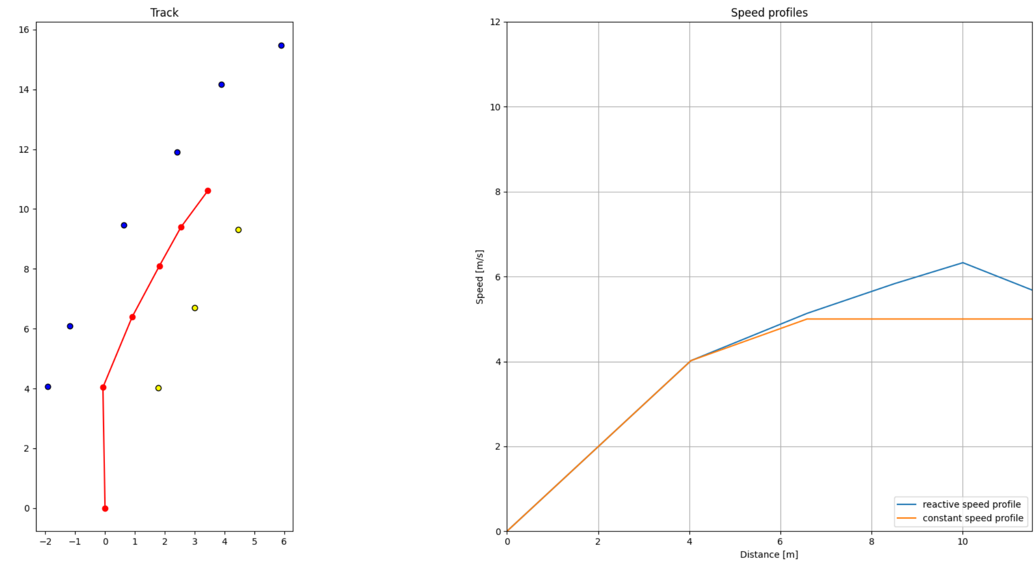

Figure 4.2: The test on the local track with a zero initial speed.

Speed Profile	Initial Speed	Average Speed	Time	Track Distance
Reactive speed profile	0.00 m/s	4.51 m/s	3.41 s	11.52 m
Constant speed profile	0.00 m/s	4.00 m/s	3.56 s	11.52 m

Table 2: The results for the local track with a zero initial speed.

The reactive speed profile is faster than constant once only in 0.15s and also there is a small difference in an average speed. But it was expected to see the small differences, because the formula was accelerating from zero speed.

Subsection 4.2.2

Local Track with a Non-Zero Initial Speed

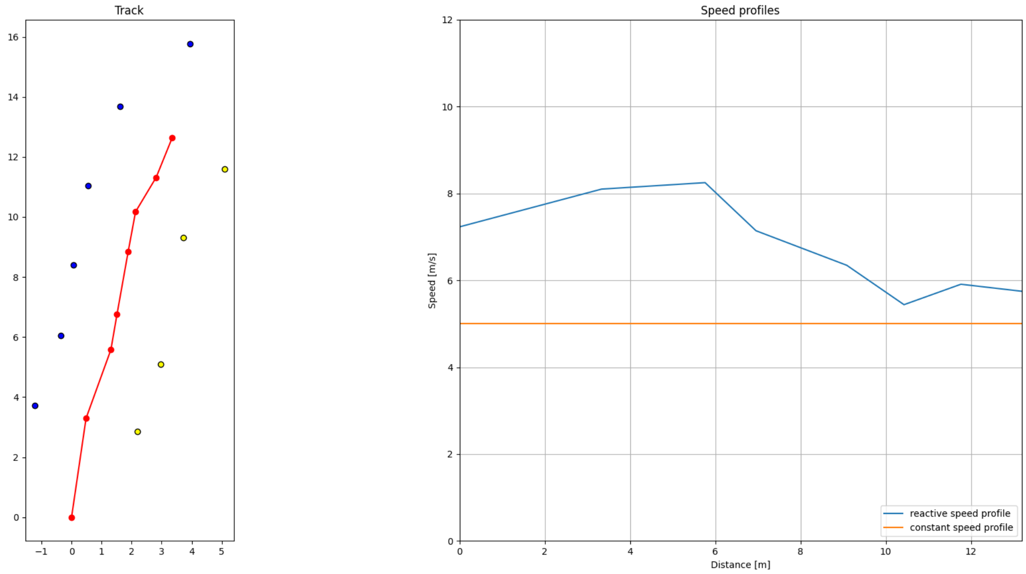

Figure 4.3: The test on the local track with a non-zero initial speed.

Speed Profile	Initial Speed	Average Speed	Time	Track Distance
Reactive speed profile	7.23 m/s	6.77 m/s	1.91 s	13.19 m
Constant speed profile	5.00 m/s	5.00 m/s	2.64 s	13.19 m

Table 3: The results for the local track with a non-zero initial speed.

In this test, the reactive and constant speed profiles have bigger differences. The reactive speed profile was in 1.4x times than constant speed profile.

Section 4.3
Global Track

To test on the global track 4.4 the SLAM algorithm was used to generate a global cone map. The start position is roughly at (0,0) and the car moves in a clockwise direction.

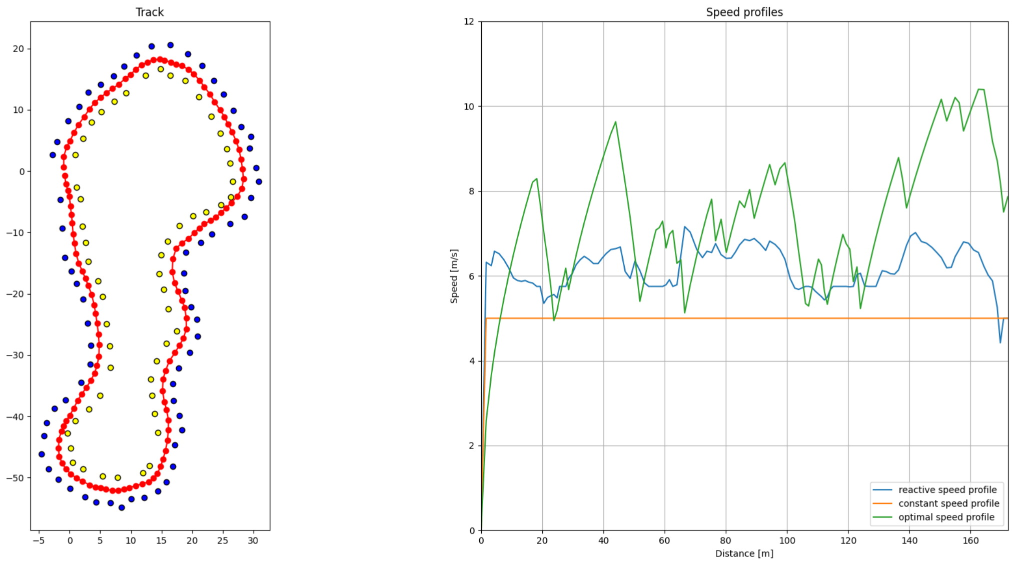

Figure 4.4: The test on the local track with a non-zero initial speed.

Speed Profile	Average Speed	Time	Track Distance
Optimal speed profile	7.21 m/s	25.37 s	172.16 m
Reactive speed profile	6.12 m/s	27.95 s	172.16 m
Constant speed profile	4.92 m/s	35.69 s	172.16 m

Table 4: Global test results.

Obviously, the optimal speed profile [5] was the fastest, because it knew perfectly the whole track. But the reactive speed profile was slower only in 2.58s than the optimal algorithm. The difference in time between the reactive and the constant speed profiles is 7.74s.

Section 4.4
Test in Simulator

The video of the test in the simulator 4.1 can be found on YouTube.

Speed Profile	Hit Cones	Time	Track Distance
Reactive speed profile	1	33 s	172 m
Constant speed profile	0	42 s	172 m

Table 5: Global test results.

The reactive speed profile showed better performance, despite one hit cone, than constant speed profile.

Chapter 5
Conclusion

This project successfully developed a speed profile algorithm that is better/faster than the algorithm with constant speed 5 m/s. In Section 1 we introduced the problem and wrote a motivation to solve the problem. After, an autonomous system and autonomous pipeline in DV.01 was presented in Section 2. In Section 3 we proposed an algorithm for the first lap: reactive path planning 3.1 and reactive speed profile 3.2. The next step was to validate the algorithm and it was made in Section 4 and the final results will be discussed in Section 5.1. The algorithm that was proposed is quite simple and there will be some improvements in the future 5.2.

Section 5.1
Achieved Results

The proposed algorithm achieved the main goal - it is faster than the constant speed profile. The algorithm was tested in different conditions. On the local track with a zero initial speed it was faster in 0.15s 2 and with a non-zero initial speed it was faster in 0.73s 3. The bigger difference was on the global track - 7.74s 4 and in the simulator test - 9s 5. It was not tested on the formula in the real world, but hopefully it will be in the near future.

Section 5.2
Future Work

The future work is a big and important part of this project. The algorithm can be improved in many different ways: use complex physical laws or complex model of a car, learn friction coefficient during the race, use spline interpolation to have smooth path, test the algorithm on more different tracks, make changes in a simulator, analyze why reactive speed profile sometimes has lower speed than constant profile, what is the optimal speed that should be set. And the main future work is to test it on DV.01 on the real race track.

Chapter A
References

[1] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[2] Roman Šıp. Visual detection of traffic cones for autonomous student formula. Bachelor's thesis, Czech Technical University in Prague, 2022.

[3] Daniel Štorc. Detection of traffic cones from lidar point clouds. Bachelor's thesis, Czech Technical University in Prague, 2022.

[4] Tomáš Roun. Navigation system for autonomous student formula. Master's thesis, Czech Technical University in Prague, 2021.

[5] Michal Horáček. Finding the fastest trajectory for autonomous student formula. Bachelor's thesis, Czech Technical University in Prague, 2022.

[6] Sebastian Thrun, Michael Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23:661–692, 1 2006.

[7] Nitin R. Kapania. *Trajectory planning and control of an autonomous race vehicle*. PhD thesis, Stanford University, 2016.

[8] Formula Student Germany. Formula student rules 2023. `https://www.formulastudent.de/fileadmin/user_upload/all/2023/rules/FS-Rules_2023_v1.1.pdf`, 2023.

[9] eForce FEE Prague Formula. Engineering design report: Aerodynamic devices. 2018.