Tartu University
Faculty of Science and Technology
Institute of Technology

Uroš Petrović

Donkey Car platform for autonomous driving research

Master's thesis (30 EAP)

Robotics and Computer Engineering

Supervisor:
Dr. Naveed Muhammad

Tartu 2023

Resümee/Abstract

Donkey Car platvorm autonoomse sõidu uurimiseks

Isejuhtivad autod (autonoomsed autod, roboautod) on sõidukid, mis on võimelised sõitma ilma inimese sekkumiseta. Viimastel aastakümnetel on see teadusvaldkond saanud palju tähelepanu. Kasvab vajadus liikluse ohutuse ja tõhususe parandamise järele. Isejuhtivad autod lubavad olla ohutud. Peamine põhjendus on see, et inimlike vigade kõrvaldamise ja algoritmide kasutamisega saaks enamikus olukordades ettetulevate vigade ja õnnetuste arvu drastiliselt vähendada. Autonoomne sõit on praegu põhjalikult uuritud valdkond, kuid olulised väljakutsed on endiselt lahtised, mistõttu on järgmistel aastakümnetel vaja teha märkimisväärseid uuringuid. Kuna elusuuruses katseplatvormide maksumus autonoomse sõidu uuringute jaoks on väga kõrge, tekib vajadus kasutada katseplatvormidena väiksemaid sõidukeid. Käesolevas lõputöös uuritakse sellise meetodi elujõulisust, hinnates väikesemahulise isejuhtivate autode platvormi Donkey Car jõudlust ja sobivust isejuhtimise uuringute erinevate aspektide jaoks. Hinnati nii isejuhtimist, lokaliseerimist, kaardistamist ning platvormi võimalusi. Saavutatud tulemused näitasid, et Donkey Car on vastuvõetav platvorm väikesemahuliste autonoomsete uuringute jaoks.

CERCS: S274 Teaduse uurimismetodoloogia, T125 Automatiseerimine, robootika, control engineering

Märksõnad: väikesemahulise autod, autonoomsed autod, SLAM, hindamine, isejuhtivad

Donkey Car platform for autonomous driving research

Self-driving cars (autonomous cars, robo cars) are vehicles that are able to drive without human input. In the recent decades, this field of science has received a lot of attention. There is a growing need to improve the safety and effectiveness of traffic. Self-driving cars have the promise of being safe. The main reasoning is that by removing human error, and utilising algorithms, the amount of mistakes and accidents that would arise in most situations could be drastically reduced. Autonomous driving is a heavily investigated area currently, but significant challenges remain open, thus requiring significant research during the decades to come. As the cost of life-size test platforms for autonomous-driving research is very high, the need to use smaller-scale vehicles to be employed as test platforms arises. This thesis investigates the viability of such a method, by evaluating the performance and suitability of a small-scale self-driving car platform, the Donkey Car, for different aspects of self-driving research. Self-driving, localization and mapping as well as the platform's capabilities were evaluated. The results achieved showed that the Donkey Car is an acceptable platform for small-scale autonomous research.

CERCS: S274 Research methodology in science, T125 Automation, robotics, control engineering

Keywords: Small-scale cars, autonomous vehicles, SLAM, evaluation, self-driving

List of figures

List of tables

Abbreviations

ROS - Robot Operating System

SLAM - Simultaneous Localization And Mapping

LIDAR - LIght Detection And Ranging

RC - Remote Car

IMU - Inertial Measurement Unit

CNN - Convolutional Neural Network

UT - University of Tartu

CAD - Computer Aided Design

RP - Rotating Platform

SBC - Single Board Computer

DIY - Do It Yourself

AV - Autonomous Vehicle

ADL - Autonomous Driving Lab

Contents

1 Introduction

Autonomous vehicles have become prevalent in the last decade, gaining commercial popularity. The knowledge surrounding this technology is still in its early stages, with many companies racing to become the best in this field. The main issues currently are dealing with safety and economic aspects.
There is a growing need to increase the testing as most vehicles rely on data to be able to successfully learn to drive. One potential method of providing easy and cheap research is by utilising small-scale self driving platforms. This would hasten the research by easy deployment of small vehicles and cut costs with budgeting. This thesis tackles the suitability of such a solution by evaluating a self-driving platform, Donkey Car, on many different aspects which would make a small-scale platform usable for autonomous research.

1.1 Contributions

The work done in this thesis aims to improve the scientific aspect of small-scale autonomous vehicle platform evaluation by:

- Introducing categories necessary for evaluation
- Creating a descriptive scoring of the categories
- Validating evaluation scores through experimentation

1.2 Structure of the manuscript

The remainder of the manuscript is structured as follows. Chapter 2 will tackle literature review on autonomous vehicles, the challenges that current research poses, and give a brief introduction to small-scale platforms. The Donkey Car platform will be covered in greater detail in chapter 3, while in chapter 4 ROS will be covered as it will be utilised in experiments. In chapter 5 preliminary work regarding the evaluation will be covered, as these results will be used in the final evaluation of the platform. Chapter 6 explains the methodology on which the Donkey Car platform will be evaluated and the experiment setup that was done. In chapter 7 the results achieved in the main experiment will be shown and in chapter 8 the discussion of the final scoring will be done. Finally, chapter 9 will give the conclusion of the thesis and provide suggestions for future work regarding this topic.

2 Literature review

This section covers the introduction to autonomous vehicles, challenges in autonomous vehicle research, methods used in self-driving cars and small-scale cars.

2.1 Autonomous vehicles

An autonomous vehicle (Figure 1) [1] is a vehicle that is able to perform all its functions without the need of any human intervention, through its ability to sense and perceive its surroundings. To do that it employs different types of sensors such as cameras, lidars, GPS, odometry, inertial measurement units (IMU) etc.
Autonomy in autonomous vehicles is divided into six levels [2], each representing different ratio between human input necessary and autonomy of the car:

- Level 0 - no automation: The vehicle is equipped with no autonomation, requiring the human driver to be fully in control of the vehicle
- Level 1 - hands on: The vehicle is able to do one or more features automatically, such as cruise control. The human driver need to perform all the other tasks
- Level 2 - hands off: Vehicle is able to perform two or more features autonomously, such as steering and acceleration. The human driver needs to monitor all tasks and perform the rest.
- Level 3 - eyes off: The vehicle has environmental detection capabilities and uses it to drive. The human can take his eyes off the road but still needs to be able to intervene should the need arise.
- Level 4 - mind off: The vehicle is able to perform all driving tasks under certain conditions. The human driver still has the option of intervention.
- Level 5 - steering wheel optional: The vehicle is able to perform all driving tasks under all conditions. No human attention or intervention is needed.

Figure 1: A Tesla autonomous vehicle driving on the road. Source: Micheal Simari, Car and Driver [3]

In order to perform driving, most autonomous vehicles use decision making using machine learning methods [4]. They usually follow a specific framework [5]:

- Input
    - Driving environment
    - Status of ego vehicle
    - Map
- Decision-making
- Output

Raw data is usually collected from different types of sensors to perceive the driving environment such as lidar, camera, radar etc. This data is processed in order to generate the environment status.

The status of ego vehicles is usually acquired using IMU or similar sensors with information from the motion estimation system.

A map can provide a lot of information down to the lane level that can be utilised as a secondary means of perception in order to increase accuracy and reduce computation power needed.

The output of the decision making system can be divided into two categories:

- High level behaviours such as lane changing, merging etc.
- Low level commands such as steering and acceleration

This system is designed to mimic human-like safe and reliable driving strategies, a challenge current autonomous systems face. It is necessary for vehicles to make these decisions with no fault and with no delay.

2.2 Challenges in autonomous driving

There are many challenges current autonomous systems face [6], some of them being:

- Safety: Current technology must be tested in great detail before being put into commercial use. The World Health Organization states that vehicle accidents are one of the leading causes of human death [7]. As the goal of autonomous vehicles is to reduce this, they need to be able to make little to no mistakes during driving.
- Technical barriers: modern technology for autonomous driving has yet to be developed enough to be able to perform in all kinds of weather, and situations. Therefore it is necessary that more research and development is made to create reliable sensors and computers to be able to accommodate different scenarios that autonomous vehicles must learn to react to properly

Other challenges, like legal challenges, integration in city transport, are as important as the rest, but are not the topic of this thesis, so they will not be addressed.

Tackling these issues is very important. In order to develop these methods, however, most researchers do not have access to a large-scale vehicle, or no financial capacity to perform it. Thus the growing need to scale down the vehicles for testing has arisen.

There are many such platforms that can facilitate autonomous driving research, such as Donkey Car [8], OpenBot [9] and GopiGo3 [10], but this thesis will focus on one such platform, namely Donkey Car, in order to assess the viability of small scale self-driving cars for autonomous research.

2.3 Small-scale car platforms

Figure 2: Small-scale platforms. From left to right: Verti-Wheelers, OpenBot, GoPiGo3

There have been many different small scale vehicle platforms developed (Figure 2), each with a different methodology of working and each having different advantages and disadvantages:

- Donkey Car [8]
- OpenBot [9]
- GoPiGo3 [10]
- Verti-Wheelers [11]

OpenBot is a platform for turning smartphones into robots, utilising the powerful hardware and sensor suite of modern smartphones as the brains of robots.
GoPiGo3 is a simple platform for teaching robotics. Housed with simple and cheap components this is a great tool for learning core robotics and autonomous functions.
Verti-Wheeler aims to solve the issue of many small scale platforms' inability to handle rough terrain.

2.4 Small scale autonomous vehicle platform research

Small-scale AVs have proven to be able to match commercial vehicles in almost all areas [12][13]. Research shows that small scale vehicles have the potential to create a low-cost, easy to integrate solution that will further the area of autonomous vehicle research. These platforms are designed to mimic commercial AVs:

- Equipped with a sensor suite necessary for AVs (lidar, camera...)
- On-Board Computer (OBC) to perform necessary calculations and processes
- Chassis to have an appearance close to a real AV and to house all components

After each platform is designed, experimentation is done to assess the viability of such a platform for the task they were designed to do (which is usually one or more areas within the broader field of autonomous-driving research).
Object detection proved to be robust and scalable on a small-scale AV. Research [12] shows that the results achieved are acceptable enough, with some small issues regarding the confidence of the detected objects. The self-driving aspects of platforms gave results acceptable for academic learning.
Self-driving aspect testing on the Go-CHART [14] platform confirmed the use of small-scale AVs to test human-driver and driver-driver situations in a safe and controlled environment.

The platform was able to perform lane following and object detection at an acceptable rate. There were issues with the inference rate of some configurations due to the hardware limitation but ultimately this platform was stated to be able to execute autonomous driving capabilities at a level acceptable enough for research.
Small scale platforms exhibit state-of-the-art localization and mapping research capabilities [12]. Through testing Lidar based SLAM methods, results achieved were sufficient for further improvements in this area of autonomous vehicle research.
Small-scale platforms also show acceptable results in specific autonomous driving concepts like parking [13]. Testing was done on different areas necessary to perform this specific task (perception, self-driving, localization…) and the results achieved were acceptable to further research in autonomous parking.
Verti Wheelers have shown promising results [11] in driving in rough terrain, and allowed for development of different control algorithms like open-loop, rule based and end-to-end driving.

2.5 Localization and mapping

Localization and mapping is one of the pillars necessary in autonomous driving. This is the ability of the car to orient itself in the real world, and its ability to create a map of the environment using various sensors that it has access to.
In order to do these tasks the most common method autonomous vehicles use is SLAM (Simultaneous localization and mapping) algorithms. This is the method of construction of a map and orientation of a vehicle simultaneously. There are many different algorithms that are used to do this, and some of them are:

- Particle filter [15]
- Kalman filter [16]
- GraphSLAM [17]

2.6 Lidar

Light Detection And Ranging (Lidar) is one of the most common sensors for autonomous vehicles. Used for localization and mapping, this device uses light in the form of a pulsed laser to measure distances. The Lidar device (Figure 3) emits pulsed light waves from a laser into the environment. These laser waves are then returned from objects they collide with and return to the sensor, where the time it took for the wave to return is calculated. The device is also rotated at very high speeds in order to create a set of points that can be processed into a map.

Figure 3: A Lidar device

2.7 Summary

Research done on many different small-scale platforms have shown that using small scale-platforms is quite cost-effective and easy to deploy. These platforms were tested on many different aspects of autonomous research (self driving, localization and mapping, parking, object detection…) and all achieved quite acceptable results. This leads to the question of more systematically evaluating how suitable a platform might be for autonomous driving research. Furthermore, this gives cause to evaluate other small-scale platforms as well, like the Donkey Car.

3 Donkey Car

Donkey Car (Figure 4) [8] is an open source DIY self-driving platform for small-scale cars. This platform allows for the custom creation of a self-driving vehicle that is able to be controlled remotely (using a smartphone or a computer) or drive on its own. The main focus of this platform is fast experimentation and easy contribution.

Figure 4: A Donkey Car. Source: Donkey Car [8]

The Donkey Car library supports different SBCs like the Raspberry Pi or Jetson Nano. Its code is written in the Python programming language.
As this is a custom made car there are a lot of different configurations hardware-wise that can be made. For this thesis the following configuration is used:

- Raspberry Pi 4B [18]
- RoboHAT MM1 robotics controller board [19]
- Raspberry Pi Camera [20]
- 1/16th 4WD Electric Power R/C Off-Road Truck [21]
- RPLidar A1 [22]

3.1 Donkey Car software architecture

The Donkey Car code is organised into parts (Figure 5) that take various inputs and return outputs. A part is a Python class that wraps a functional component of a vehicle (such as sensors, actuators, pilots, etc.) Each part is constructed and then added into a vehicle loop.

```python
class DonkeyCarPart:
    def run(self, input_variable):
        output_variable = proecess(input_variable)
        print(f"{output_variable}")
        return output_variable

# create the vehicle and it's internal memory
V = dk.Vehicle()

# initialize a variable in the vehicle's memory
V.mem['input_variable'] = 10

# add the part to the vehicle.
V.add(DonkeyCarPart, inputs=['input_variable'], outputs=['output_variable'])

# start the vehicle loop
V.start()
```

Figure 5: An example Donkey Car script that executes a part

Donkey Car comes with a premade template called manage.py which contains the main code to run the car. This code is run in a vehicle loop that will be executed at a rate specified by the DRIVE_LOOP_HZ value. Most of these values are stored in myconfig.py and config.py for ease of editing. There are also a couple of concepts that are important in the Donkey Car template:

- Memory: this is a hash map of all various vehicle values that are shared by all parts, such as inputs, outputs and conditions
- Inputs: these are memory values passed to the run() method of any part. Every time the run() method is called, the vehicle loop will take the input from the memory and pass it to the run() method of the part for execution
- Outputs: these are memory values that will be returned by the run() method. These methods are written into the vehicle memory after each loop is executed
- Run_condition: this is a memory value that can be used in order to decide if a part should be executed or not. This is the way that a part can be turned on or off during execution.
- Run_threaded: this is a method similar to the run() method that is used if the need for some processes to execute faster than the vehicle loop are needed. Unlike the run() method, this method does not require any inputs. However, all threaded parts require an update() method that will run separately from the vehicle loop in order to achieve execution as fast as the python's scheduler will allow

3.2 Autopilots

Donkey supports deep-learning, path following and computer vision for autopilots.
The deep-learning autopilot uses a single forward facing camera and convolutional neural networks (CNN) to train an autopilot using Imitation learning (Behavioural learning). This technique has its name from its goal, and that is to imitate the actions of a human. The process of training a deep learning autopilot goes as follows:

- A human driving needs to collect data. While driving the Donkey Car collects records at 20 times per second. The record contains 3 components: a camera image, the throttle value, and the steering value. In order to have a successful model the recommendation is to collect around 10000 images.
- After the data is collected, some editing needs to be done to remove any crashes or mistakes. Donkey Car has an excellent utility for this - Donkey UI where easy clean-up of data can be done.
- When the data is edited, and there is a satisfactory amount of images, this data is then used to train a CNN model
- After the Network is trained, it is used to infer throttle and steering values given an input image

The path following autopilot allows for the Donkey Car to use GPS in order to follow a specified path. The process of training a path follow autopilot is:

- The car needs to be driven by a human in order to gather data. Each record represents the 2D coordinates of the car's position in the world in metres.
- The autopilot uses all records of the path and adjusts its steering and throttle in order to reach each waypoint and follow the path

The computer vision template is similar to the deep-learning platform in data acquisition. The difference is that it uses more traditional computer vision algorithms to train. The most common use of this autopilot is for line following.

3.3 Deep learning

The deep learning template uses Keras models with Tensorflow backend. These models are customizable, however Donkey Car has several models that they recommend using:

- Keras Categorical
- Keras Linear
- Keras IMU
- Keras Latent
- Keras RNN
- Keras 3D
- Keras Behavior
- Keras Localizer

These models all follow a similar architecture. The most common model, the Linear model, is used for low-compute environments, has five convolution layers followed by two dense layers. The disadvantages of such a model is that it might have difficulty learning to throttle well.

4 ROS

Robot operating system (ROS) [23] is an open-source set of libraries and tools for use in robot applications. It is one of the most common methods of robotics simulation and research. This is also quite common in larger autonomous vehicles to employ its use for efficient processing. As ROS is quite extensive only the concepts that will be used in the thesis work will be covered in this chapter.

4.1 Architecture

ROS is specifically designed for modularity and multi purpose use by employing the use of workspaces. A workspace is a folder containing ROS packages. A ROS package is the smallest unit that can be built in ROS, housing a specific set of files in folders that can execute or solve a specific task or problem.

4.2 Nodes

Nodes are processes that are responsible for computations. Instead of using a single file or script that will be doing everything, ROS has nodes that are working to execute smaller functions during execution.

4.3 Publishers and subscribers

Publishers and Subscribers are one of the most important concepts of communication in ROS. These are nodes that are responsible for publishing and receiving messages that are being used in the process of a simulation.
The publisher is responsible for sending (publishing) messages in the ROS network. Each publisher has specified the message type it will send, and to what topic the message will be sent to.
The subscriber is responsible for receiving messages. Each subscriber has to have specified what message type and the topic it is receiving, and the callback function it should execute once a message of this type is received.

4.4 Rviz

Rviz [24] is a visualisation tool for ROS. This tool proves invaluable for simulation purposes and visualising complex tasks and sensor outputs. For most robotics tasks, a simulation is done before moving on to practical experiments.

4.5 Hector SLAM

Hector SLAM [25] is a ROS package providing a localization and mapping system. This package can be used without odometry, as well on platforms that have sensors with roll/pitch motions. It utilises the fast update rate of modern Lidar systems to estimate the 2D pose of the sensor.

4.6 Cartographer

Cartographer [26] is a system that provides real-time SLAM in 2D and 3D systems. It utilises two subsystems: a local and global SLAM. The local SLAM is responsible for creating a succession of submaps. The global SLAM is responsible for finding loop closure constraints by scanning matching scans against submaps. This is a C++ library, however, the developers have provided ROS integration for fast deployment.

5 Preliminary work

Some aspects of evaluation have been previously done at the University of Tartu under various courses. There are two such experiments: the indoor and outdoor aspects of autonomous driving . The indoor and outdoor aspects were explored during the Neural Networks course [27], and indoor aspects during the Machine learning course [28].

Outdoor evaluation was done in two experiments. The first experiment was exploring the performance of the Donkey Car in various road conditions and locations in Tallinn (Kopli Cemetery Park, Russalka Monument, Majaka street). These experiments have shown that the car was having trouble adapting to the different road types, discharging the battery faster on the rougher roads. The outdoor model training produced acceptable results at locations where the track was more defined (Majaka street and Kopli Cemetery Park asphalted road).

Indoor evaluation was done in two experiments at the Delta building of the University of Tartu: one was evaluating Donkey Car's ability to learn to drive in a toy city (cf. Figure 6), and the other was giving way to other vehicles using right-side priority (cf. Figure 7).
To evaluate the ability to drive in the toy city, data was recorded during different configurations of lighting (day and night) and different locations of obstacles. Multiple model types were then trained (Linear, RNN, 3D) using different configurations of data (only day, only night, combined). The car was able to drive around the track and perform very basic obstacle avoidance, and good steering and self-driving capabilities. There were hardware issues that caused the car to not throttle well caused by lower battery levels over short periods of time (20 mins), but these issues were able to be fixed by forcing the car to drive at a fixed speed when the battery was full. The Linear model performed the best, with the other models having issues in reacting to the environment due to low compute power of the Raspberry Pi. All of these issues and results matched with the documentation provided by the Donkey Car developers [29].

Figure 6: Indoor experimentation of self-driving capabilities of Donkey Car

To evaluate the ability to detect other vehicles and allow them a right-side priority, a track in the shape of digit “8” was created, and using two Donkey Cars, data was collected at many different scenarios for giving the right-side priority (car far away, car on the left, car on the right…). The model was trained using the Linear and RNN model. The Linear model reported to have faster response time (~20ms), but the RNN was too slow to react (~400ms) which is unacceptable for AV.

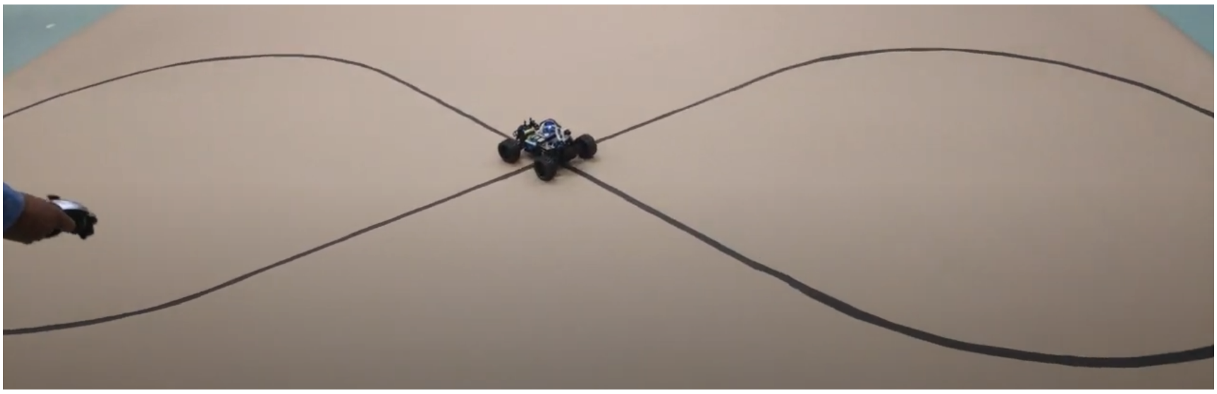

Figure 7: Second indoor experiment setup of the digit 8 track

The experiment results achieved in these preliminary works confirmed that the Donkey Car is suitable for self-driving research on a small scale. For more complex situations, better hardware is needed for testing.

6 Methodology

To evaluate suitability of a specific aspect of any platform, the need for setting categories is important. As the main question addressed in this thesis is research using small-scale platforms, there are several categories one such platform will be evaluated on:

1. Autonomous driving capabilities
2. Sensor integration
3. Scalability
4. Adaptability
5. Integration with other tools
6. Documentation and support
7. Performance
8. Cost
9. User experience

These categories were selected through literature review (presented in chapter 2) and personal experience with hardware platforms.
The categories are described below, followed by an evaluation scheme and the experimental setup used in this study.

6.1 Autonomous driving capabilities

This is the capability of the platform to perform simple or more complex autonomous driving tasks. In order to evaluate this category, experiments testing obstacle detection and avoidance, control and localization and mapping should be done. Using the Donkey Car deep-learning template, the capability of learning to drive indoors and outdoors and its control was evaluated in preliminary work presented in chapter 5, while using ROS the localization and mapping capabilities were explored, and is presented further in chapter 7.

6.2 Sensor integration

This is the quality of the sensors the platform has and its potential for improvement. As the hardware of the Donkey Car is customizable, this will be evaluated on the current sensor configuration by exploring ease of setup and integration. These sensors will be evaluated alongside the autonomous capabilities, as the deep-learning template utilises the camera, and ROS will utilise the Lidar and IMU (Inertial Measurement Unit).

6.3 Scalability

This is the potential of the platform to be scaled to different types or sizes of autonomous vehicles, and the ability to scale by complexity as well. This category will be evaluated during the main experiment and the previous work result presented in chapter 5.

6.4 Adaptability

This is the ability of the platform to handle different environmental conditions such as different roads types and weather and lighting conditions. This was evaluated through indoor and outdoor testing presented in chapter 5.

6.5 Integration with other tools

This is the potential of the platform to integrate with other frameworks and tools in autonomous driving research. As autonomous vehicles are also called “robo cars”, ROS will be the main tool the evaluation will be performed on. During the localization and mapping experiment, SLAM methods will be evaluated and explored, whether or not the integration on the Donkey Car platform was seamless.

6.6 Documentation and support

This is the quality and coverage of the platform in the form of documentation and support resources. This will be evaluated by exploring the documentation provided by the developers of Donkey Car, and the support given to users experiencing issues.

6.7 Cost

This is the overall pricing of the platform. This is evaluated by comparing the average price of a commercial autonomous vehicle (like the Tesla car), the price of an autonomous vehicle setup for research (like the Lexus RX450h vehicle available at the ADL in UT) with the average price of various small-scale vehicle platforms.

6.8 User experience

This is the platform’s user experience design. This will be evaluated by exploring the Donkey Car software and difficulty of setup during experimentation and review.

6.9 Evaluation criteria

Every criteria will be evaluated using a 3-point descriptive assessment:

- 0 - insufficient: The platform insufficiently addresses the requirements for a specific category
- 1 - sufficient: The category meets the minimum requirements of successful research capabilities for academic purposes (university studies, training)
- 2 - promising: The category meets the requirement of successful research applications for both academic and commercial purposes with the potential for further improvement

To evaluate a category, experimentation and research must be done. Each scoring needs to have a reasoning (justification) on why this scoring is given. The aim of the evaluation is to be as objective as possible, however it needs to be stated that it is near impossible to not employ subjective opinions during any kind of evaluation. Therefore scoring should reflect the impartialness as much as possible.

6.10 Experiment setup

To evaluate the localization and mapping aspect of the Donkey Car platform, ROS Noetic is used. As the hardware of the car is not powerful enough to handle heavy computing, communication between the car and a more powerful computer is necessary, hence the use of ROS. Its publish and subscribe methods are excellent in evaluating both the sensors and the localization methods.

ROS has many different localization and mapping packages, however, two will be evaluated in the current study: Hector SLAM [6] and Cartographer [7]. The location chosen for this evaluation is the University of Tartu (UT) Delta building, specifically the second and third floors. UT has provided its students a CAD drawing of each floor, giving a perfect comparison for the results acquired.

The SLAM packages are designed for use with limited sensors, so a Lidar is set up on the Donkey Car system. Donkey Car currently supports the RPLidar series of Lidars, so the RPLidar A1 Lidar [22] will be used.

To evaluate the platform's capabilities there will be two scenarios: a small area of the second floor (Figure 7) will be mapped, and a larger area of the third floor will be mapped. The small section will be run until an acceptable result is achieved, with high accuracy to the comparison point. The larger section will evaluate the platform's capabilities of running during longer periods of time and its performance.

Figure 8: Experiment setup of the second floor. The Donkey Car is placed in front of room 2018 of the Delta building

6.10.1 Categories evaluated using the experimental setup

The following categories will be covered by the above-mentioned experimentation:
- Autonomous capabilities
- Sensor integration
- Scalability
- Adaptability
- Integration with other tools
- Performance

7 Results

Mapping a small section of the second floor resulted in acceptable results (Figure 8). There were issues setting up the software part of the experiments due to the hardware - the Donkey Car was having issues communicating with the Inter-Integrated Circuit (I^2C). After troubleshooting the issue was fixed and the Hector SLAM was able to produce a similar result as the section in the CAD drawing. Due to the high compute requirements of the SLAM method and the low compute power of the Raspberry Pi, all calculations were offloaded to a computer. Hector SLAM is able to use only lidar data to create a map, however this meant that the car needed to drive quite slow to produce acceptable results.
The mapping of the third floor provided acceptable results with better integration and setup as this experiment was done after the first one. There were issues mapping long corridors where the car was unable to orient and localise itself correctly (Figure 9b). These issues are solvable by using better hardware. Hetcor SLAM had much more trouble localising then the Cartographer (Figure 9a), however, comparing with the CAD drawing of the floor (Figure 9c) we can say that the results achieved are acceptable enough.

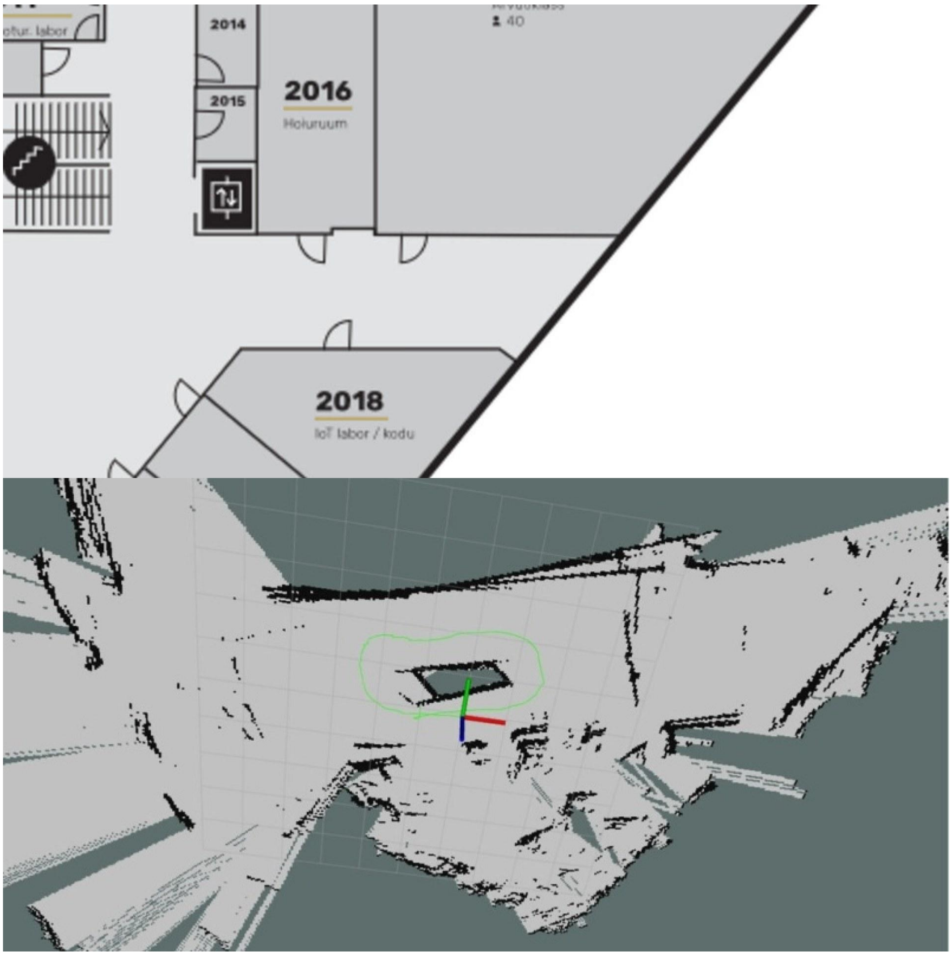

Figure 9: Results of Hector SLAM in front of room 2018 of Delta building with the CAD drawing of the section mapped

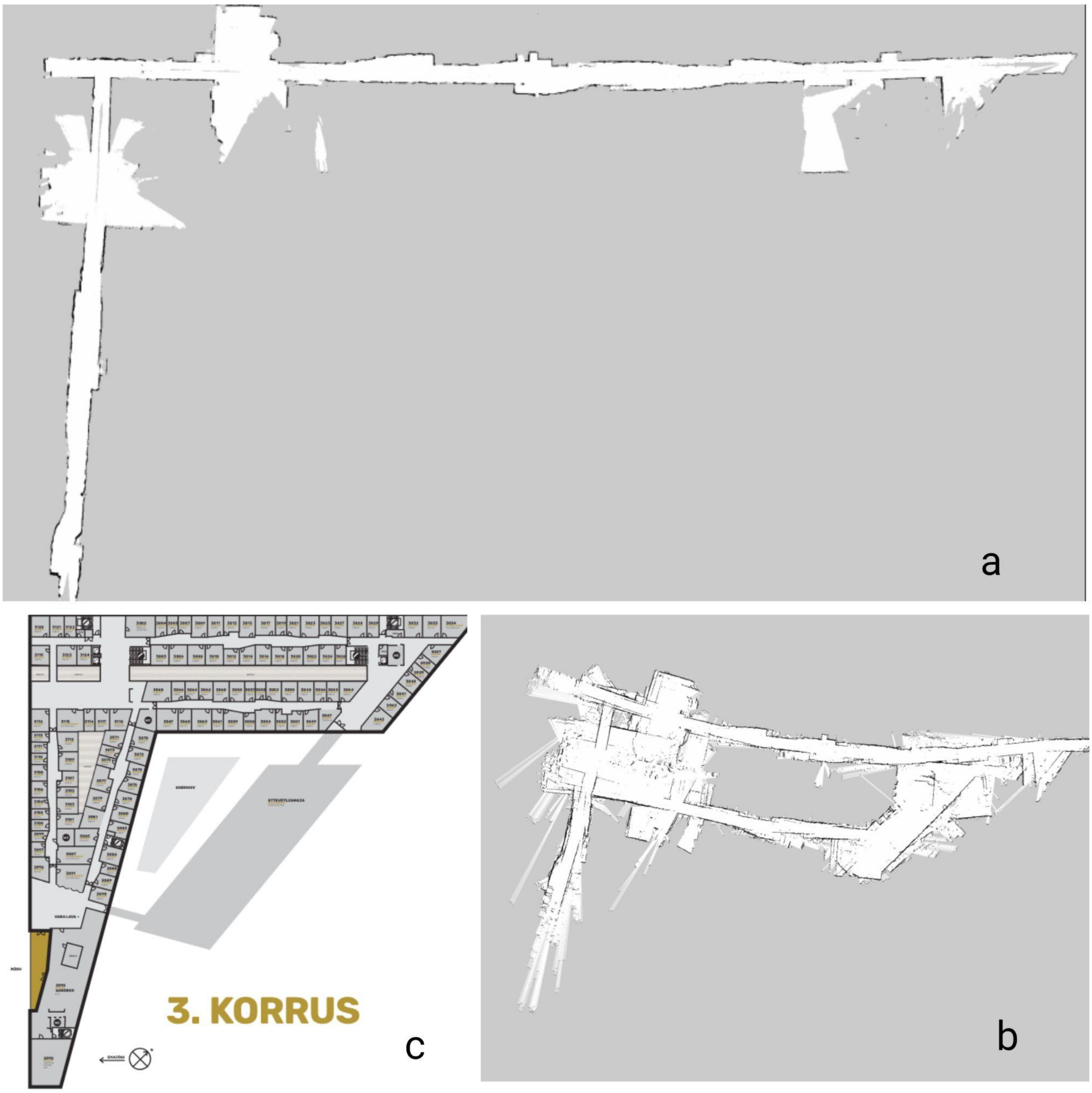

Figure 10: Experiment results on the 3rd floor of Delta building. a) Cartographer results b) Hector SLAM c) CAD drawing of the 3rd floor

Most of the hardware issues that were presented in chapter 5 appeared in this experiment, for example the slowdown of the car due to the battery level lowering. This issue was on a lesser scale, as the car was slowed down from the start, causing the battery to discharge slower than usual.

7.1 Categories evaluated

The main category evaluated in this experiment was the localization and mapping aspect of the autonomous capabilities, with several other categories being evaluated during the setup and experimentation:

- Sensor integration: integration of Lidar and IMU was met with some difficulties, but after some troubleshooting worked as intended
- Scalability: Evaluating different SLAM packages proved that scaling to different methods and processes in sufficient
- Adaptability: Evaluation of different locations (floors) of the building proved the platform is able to function with small changes to the environment
- Integration with other tools: ROS performed well on the platform, proving good tool integration capabilities
- Performance: The performance of the platform after offloading computing to another device provided the vehicle performance to increase considerably

8 Analysis and discussion

This section discusses the scoring of each category the Donkey Car platform was evaluated on (cf. Table 1), providing the reasoning and explanation on how the category was evaluated if it was not explained previously.

Category	Score
Autonomous capabilities	sufficient
Sensor integration	sufficient
Scalability	sufficient
Adaptability	sufficient
Integration with other tools	promising
Documentation and support	promising
Performance	sufficient
Cost	promising
User experience	promising

Table 1: Final Scoring of evaluation categories of the Donkey Car platform

8.1 Autonomous capabilities

Regarding self-driving aspects, the platform has difficulty dealing with more complex tasks like object detection, and obstacle avoidance [28]. There is minimal capability of training and executing simple deep-learning models. However when trying to create more complex models, there is room for improvement. The models that were covered during self-driving experiments had trouble dealing with the safety aspect, with them sometimes crashing unless the vehicle is slow, which would require using a more powerful vehicle.
As the experiments for SLAM methods are successful, this platform proves well suited for such research. The problems arising in the experiments (hallway problem), allow for more research topics for localization research under limited conditions which might prove useful for larger vehicles.

8.2 Sensor integration

Integrating sensors to the hardware posed an issue. The fault was caused due to hardware issues which when fixed allowed for subsequent sensor integration easier. This is an acceptable result, as the sensors are able to be integrated with a bit of work.

8.3 Scalability

An aspect important for autonomous research, and the reasoning for employing small-scale cars, this category is difficult to rank. While the localization and mapping and sensor suite provide enough potential for research, the current state of the self-driving aspect does not provide enough for use in larger vehicles. The Donkey Car architecture allows for easy modification and creation of custom parts and models, allowing for the issue to be solvable. The customizable aspect has its drawbacks, as some hardware has problems integrating in the code (as seen in Chapters 5 and 7).

8.4 Adaptability

During the indoor and outdoor experimentation [27][28] it was difficult for the platform to handle ever changing lighting conditions, battery issues and terrain changes. These issues have the potential of being solved by utilising more powerful equipment/hardware. However, this would increase the cost and potentially the size of the vehicle, which would make the reasoning of using small-scale vehicles obsolete.

8.5 Integration with other tools

Hardware played a crucial role in this category, as the Raspberry Pi has a plethora of libraries for multiple different situations and problems. The Donkey Car architecture itself worked well in ROS during the main experiment. The setup itself was not easy to perform, but more experienced users will be able to integrate quite successfully.

8.6 Documentation and support

The documentation of the Donkey Car library is extensive. The open source aspect provides a valuable asset for development. The developers of Donkey Car have a forum and community [30] in the messaging platform Discord, where most issues are being addressed and helped by the developers themselves or other users.

8.7 Performance

During self-driving experimentation, the main bottleneck was the hardware configuration, which seriously affected the performance of any self-driving models. These models needed to be simple as anything more complex would seriously affect the performance of the vehicle. During SLAM experimentation all computing was done on a computer as hardware of the Donkey Car is unable to handle heavy computing. While these issues are an important weakness of the platform, ultimately the results achieved were acceptable.

8.8 Cost

Vehicle	Pricing
Commercial vehicle (Tesla)	35,000- 100,000€
Research vehicle setup (Lexus RX450h or similar)	Above 100,000€
Recommended Donkey Car hardware	250€
Donkey Car used in experiments	350€
GoPiGo3	200€
OpenBot	450-700€ (without smartphone)
Verti-Wheeler	Above 1000€

Table 2. Comparison of different pricing of self-driving vehicles

This category was evaluated by comparing the recommended donkey car price, the price of the car used in experiments with large autonomous vehicles, both commercial and research, and other small-scale platforms. For one self-driving research car one could afford around 250 or more small scale cars, with potential for different hardware configurations which provides for more quality testing.

8.9 User experience

The Donkey Car software package is both beginner and advanced user friendly. There are many tools that allow for starting autonomous research with minimal knowledge in the field (Donkey Car Controller mobile application, the Donkey UI) that allow for most processes to be done with a click of a button. The open source aspect and modularity of the library allows for advanced users to integrate their code easier.

9 Conclusions

This thesis tackled the evaluation of the self-driving Donkey Car platform for autonomous-driving research. Many different aspects, that are important in terms of a small-scale platform's suitability for autonomous-driving research, were covered in the study, and evaluated through multiple experiments and literature review. Some conclusions drawn during the study are as follows.

The self-driving aspects and the localization aspects were met with sufficient results, with localization and mapping having more potential than self-driving. Adaptability, performance and sensor integration was also acceptable, by using better hardware getting huge improvements. Integration with other tools, documentation and user experience proved promising, allowing easy modification and troubleshooting of most issues. Finally the cost of the platform was promising as well, cutting cost by a considerable amount.

The results of all experiments and reviewing show the Donkey Car as an acceptable platform for small-scale autonomous research.

9.1 Future work

As this is just one of many different self-driving platforms available, it would be beneficial to apply this methodology to test other platforms, or create a universal methodology for testing self-driving platforms. The Donkey Car platform allows for easy deployment, so some work could be done to implement this on commercial autonomous vehicles. Another aspect that can be researched is finding a better hardware configuration that will allow for the platform to perform at optimal capacity.

Acknowledgements

I would like to dedicate this thesis to my sister Marija Sponza, as writing this thesis would not be possible without her invaluable support and belief in me. I would also like to thank my brother-in-law Mateja Sponza, my mother Nada Petrović and father Miroslav Petrović for their continued support. I would also like to thank my supervisor Naveed Muhammad for the support and help during the writing and thesis work.

Uroš Petrović

References

[1] E. Yurtsever, J. Lambert, A. Carballo and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," in IEEE Access, vol. 8, pp. 58443-58469, 2020, doi: 10.1109/ACCESS.2020.2983149. Draft accessible online: https://ieeexplore.ieee.org/abstract/document/9046805

[2] SAE Levels of Driving Automation Refined for Clarity and International Audience 2021). https://www.sae.org/blog/sae-j3016-update 18.05.2023.

[3] Connor Hoffman. "How Capable Is Tesla's Autopilot Driver-Assist System? We Put It to the Test" 2021. https://www.caranddriver.com/news/a35839385/tesla-autopilot-full-self-driving-autonomous-capabilities-tested-explained/ 18.05.2023.

[4] Tampuu, Ardi & Matiisen, Tambet & Semikin, Maksym & Fishman, Dmytro & Muhammad, Naveed. (2020). A Survey of End-to-End Driving: Architectures and Training Methods. IEEE Transactions on Neural Networks and Learning Systems. PP. 1-21. 10.1109/TNNLS.2020.3043505. PDF: https://arxiv.org/pdf/2003.06404.pdf

[5] Gergely Debreczeni. "How self driving cars understand the world" 2019 https://medium.com/aimotive/how-self-driving-cars-understand-the-world-map-2afac13582f9 18.05.2023

[6] B Hettige, RGS Deemantha. "Autonomous Car: Current Issues, Challenges and Solution: A Review". Presented at the 15th International Research Conference. Draft accessible online: https://www.researchgate.net/publication/366986201_Autonomous_Car_Current_Issues_Challenges_and_Solution_A_Review

[7] World Health Organization. "Road traffic injuries" https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries 18.05.2023

[8] Donkey Car official website. https://www.donkeycar.com/ 18.05.2023

[9] OpenBot official website. https://www.openbot.org/ 18.05.2023

[10] GoPiGo official website. https://gopigo.io/ 18.05.2023

[11] Datar, Aniket & Pan, Chenhui & Nazeri, Mohammad & Xiao, Xuesu. (2023). Toward Wheeled Mobility on Vertically Challenging Terrain: Platforms, Datasets, and Algorithms. https://cs.gmu.edu/~xiao/Research/Verti-Wheelers/

[12] Vincke, Bastien, Sergio Rodriguez Florez, and Pascal Aubert. 2021. "An Open-Source Scale Model Platform for Teaching Autonomous Vehicle Technologies" Sensors 21, no. 11: 3850. https://doi.org/10.3390/s21113850

[13] T. Yu, W. Lu, Y. Luo, C. Niu and W. Wu, "Design and Implementation of a Small-scale Autonomous Vehicle for Autonomous Parking," 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 2021, pp. 398-402, doi: 10.1109/CACRE52464.2021.9501311. Draft accessible online: https://ieeexplore.ieee.org/document/9501311

[14] Kannapiran, Shenbagaraj & Berman, Spring. (2020). Go-CHART: A miniature remotely accessible self-driving car robot. 10.1109/IROS45743.2020.9341770. Draft accessible online: https://www.researchgate.net/publication/345339459_Go-CHART_A_miniature_remotely_accessible_self-driving_car_robot

[15] Yatim, Norhidayah & Buniyamin, Norlida. (2015). Particle filter in simultaneous localization and mapping (Slam) using differential drive mobile robot. Jurnal Teknologi. 77. 10.11113/jt.v77.6557. Draft accessible online: https://www.researchgate.net/publication/286763834_Particle_filter_in_simultaneous_localization_and_mapping_Slam_using_differential_drive_mobile_robot

[16] Inam Ullah, Xin Su, Xuewu Zhang, Dongmin Choi, "Simultaneous Localization and Mapping Based on Kalman Filter and Extended Kalman Filter", Wireless Communications and Mobile Computing, vol. 2020, Article ID 2138643, 12 pages, 2020. https://doi.org/10.1155/2020/2138643

[17] Thrun, Sebastian & Montemerlo, Michael. (2006). The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. I. J. Robotic Res.. 25. 403-429. 10.1177/0278364906065387. PDF: http://robots.stanford.edu/papers/thrun.graphslam.pdf

[18] Raspberry Pi 4B documentation. PDF: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf 18.05.2023

[19] RoboHAT MM1 documentation. https://robohatmm1-docs.readthedocs.io/en/latest/ 18.05.2023

[20] Raspi Camera documentation: PDF: https://datasheets.raspberrypi.com/camera/camera-module-3-product-brief.pdf

[21] 1/16 wheel off road truck product description. https://www.amazon.com/YONGSHUO-High-Speed-RC-Cars/dp/B09LQ8T9NN?th=1 18.05.2023.

[22] RPLidar A1 documentation. PDF:
https://bucket-download.slamtec.com/e680b4e2d99c4349c019553820904f28c7e6ec32/LM108_SLAMTEC_rplidarkit_usermaunal_A1M8_v1.0_en.pdf 18.05.2023.

[23] Robot Operating System official website. https://www.ros.org/ 18.05.2023.

[24] Rviz official documentation. http://wiki.ros.org/rviz 18.05.2023.

[25] Hector SLAM ROS documentation. http://wiki.ros.org/hector_slam 18.05.2023.

[26] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271-1278, doi: 10.1109/ICRA.2016.7487258. Draft accessible online: https://research.google/pubs/pub45466/

[27] Karina Rao, Kwasi Akuamoah Boateng, Peep Kolberg, Tõnis Hendrik Hlebnikov, Uroš Petrović. "Self-driving Donkey Car training" 2022.
https://medium.com/@donkeycar/self-driving-donkey-car-training-7b94580eb3e2
18.05.2023.

[28] Farnaz Baksh, Debasis Kumar, Uroš Petrović, Matevž Borjan Zorec. "Teaching 1:10 scale cars to drive in a city" 2023. https://shorturl.at/aMOP9 18.05.2023.

[29] Donkey Car Keras parts documentation. https://docs.donkeycar.com/parts/keras/
18.05.2023.

[30] Donkey Car Community section https://www.donkeycar.com/community 18.05.2023.

Non-exclusive licence

I, Uroš Petrović

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   “Donkey Car platform for autonomous driving research”

   supervised by Naveed Muhammad

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Uroš Petrović
18.05.2023