

УДК 62529

DOI DOI: 10.22213/2410-9304-2021-3-95-104

## Exploiting Machine Learning for Vision and Motion Planning of Autonomous Vehicles Navigation

Rutendo Midzi,  
M. Aiman Al Akkad,  
Kalashnikov Izhevsk State Technical University

*The object of this paper is to create a system that can control any vehicle in any gaming environment to simulate, study, experiment and improve how self-driving vehicles operate. It is to be taken as the bases for future work on autonomous vehicles with real hardware devices. The long-term goal is to eliminate human error. Perception, localisation, planning and control subsystems were developed. LiDAR and RADAR sensors were used in addition to a normal web Camera. After getting information from the perception module, the system will be able to localise where the vehicle is, then the planning module is used to plan to which location the vehicle will move, using localisation module data to draw up the best path to use. After knowing the best path, the system will control the vehicle to move autonomously without human help. As a controller a Proportional Integral Derivative PID controller was used. Python programming language, computer vision, and machine learning were used in developing the system, where the only hardware required is a computer with a GPU and powerful graphical card that can run a game which has a vehicle, roads with lane lines and a map of the road. The developed system is intended to be a good tool in conducting experiments for achieving reliable autonomous vehicle navigation.*

**Keywords:** autonomous vehicles, perception, control, localisation, navigation, motion planning.

### Introduction

Autonomous vehicles refers to driverless vehicles or self-driving vehicles that can drive themselves from a starting point to a predetermined destination in “autopilot” mode. This implies using various in-vehicle technologies and sensors in the absence of a human operator like adaptive cruise control, active steering (steer by wire), anti-lock braking systems (brake by wire), GPS navigation technology, lasers and radar. There are 6 levels of vehicle automation [1]. These are: Zero automation, Driver-assistance, Partial automation, Conditional automation, High automation and Full automation. Zero automation is the most common and current model of vehicles we have on our streets. No bells and whistles. Just a vehicle with cruise control to help with long-distance driving and minimize the risk of a speeding ticket from a lead foot. Most vehicles today offer *level 0* autonomous technology [2]. Driver-assistance is *level 1*. It is an adaptive cruise control and lane-keep assist system. Adaptive cruise control will keep a safe-distance between you and the vehicle ahead by use of radars and/or cameras. It will automatically apply brakes when traffic slows, and resume speed when traffic clears [3]. Lane keep assist will help move the vehicle back into the lane should you veer off a bit. Both these systems still require the driver to be in control. *Level 1* autonomy in most new model vehicles today. Partial

Automation *level 2* is a level of automation that can assist in controlling speed and steering. The driver should always have hands on the steering wheel and ready to take control. It helps maintain the distance between the vehicle and the vehicle in front by providing steering assistance. Conditional Automation *level 3*, is where the blurring starts the line between present technology and technology that’s soon to come. *Level 3* autonomous vehicles are capable of driving themselves, but only under ideal conditions and with limitations, such as limited-access divided highways at a particular speed. Drivers are still required behind the wheel. High Automation *level 4*, autonomous vehicles can drive themselves without human interactions (besides entering your destination) but will be restricted to known use cases. Full Automation *level 5* is the true driverless vehicle. Vehicles should be able to monitor and manoeuvre through all road conditions without human control. Many of the technological components exist for an artificially intelligent vehicle today but due to regulations and legal battles, *level 5* vehicles are still many years away [4].

### Development of the system

The system is to improve scientific research on the concept of autonomous vehicles. This approach is effective as it is low cost, therefore if applied well, anyone with a computer can conduct experiments. We will be using gaming environ-

ments with the capability to drive a vehicle. The goal is to have the game vehicle drive itself as if in real life. As with all automation, the main problem faced in any sphere of study is “human error”. Automation of systems tries to eliminate this problem. In autonomous vehicles, eliminating human error reduces the number of accidents on the roads. Autonomous vehicles will also reduce road traffic. Traffic jams have been studied to mostly result from human nature. That is, our reactions as people are different. Therefore the lack of unanimity of reaction when the traffic light turns green results in some drivers having to be stuck in the middle of the road when the traffic lights turn red. Therefore, the goal is to eliminate human error and get to *level 5* of autonomous vehicles [5]. The ideal solution would be to build a vehicle to test it on the road. But due to expenses involved, a better solution would be to realise these concepts in a gaming world that has already been created. Because the methods and concepts being tested are the same, the game used does not matter [6]. The system is a self-driving autonomous vehicle. The main concepts to be reviewed are perception, localisation, planning and control. These in turn are the four subsystems available. Due to the fact that

in a gaming environment it is impossible to evaluate sensors, we will not delve into the topic of sensors. Therefore this system will only be as complex as possible in the gaming environment. The current model use sensors such as the Light Detection and Ranging LiDAR, a normal Camera, the Radio Detection and Ranging RADAR.

In order to have a fully functional system, the system has four crucial systems as mentioned earlier. These are perception, localisation, planning and control. Perception is the first. The other modules totally depend on this module. The next module will be localisation. The system will only be able to localise after getting information from the perception module. After being able to localise where the vehicle is, the next step is to plan where to go next. This is the planning module. It will use the data from the Localisation module to draw up the best path to use. After knowing the best path to use, the system will have to control the vehicle without anyone helping with the control. Fig. 1 will show the structural scheme of the system. The system has 4 subsystems. Each subsystem has a module or several ones. Fig. 2 will show how the subsystems are connect to each other and their direct dependencies.

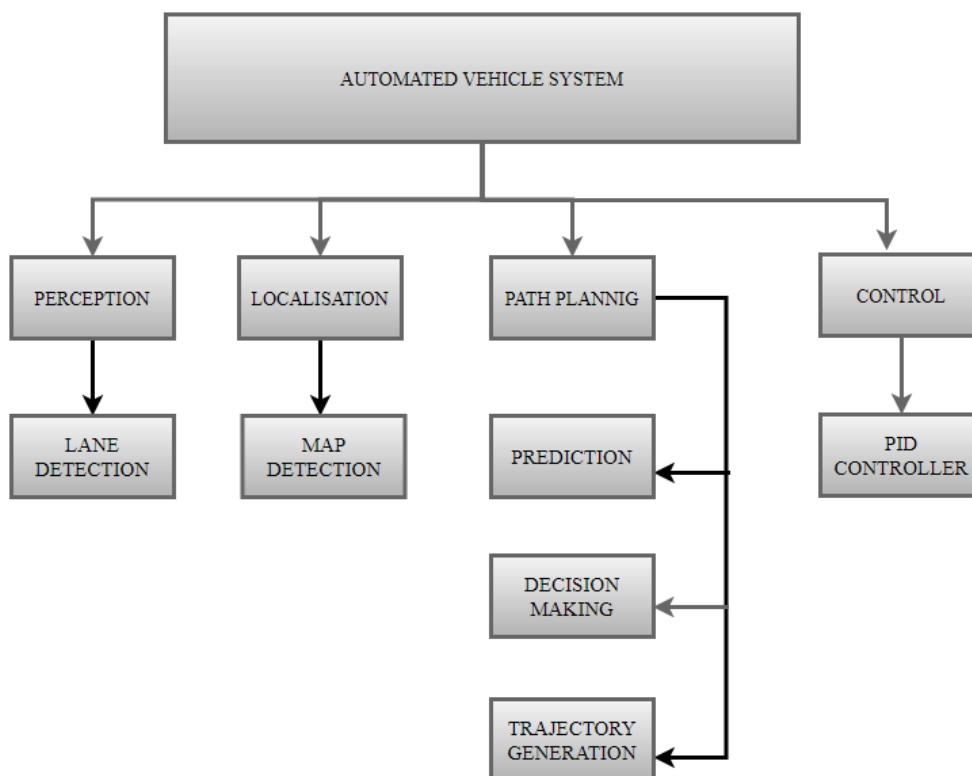


Fig. 1. Structural scheme  
Рис. 1. Структурная схема

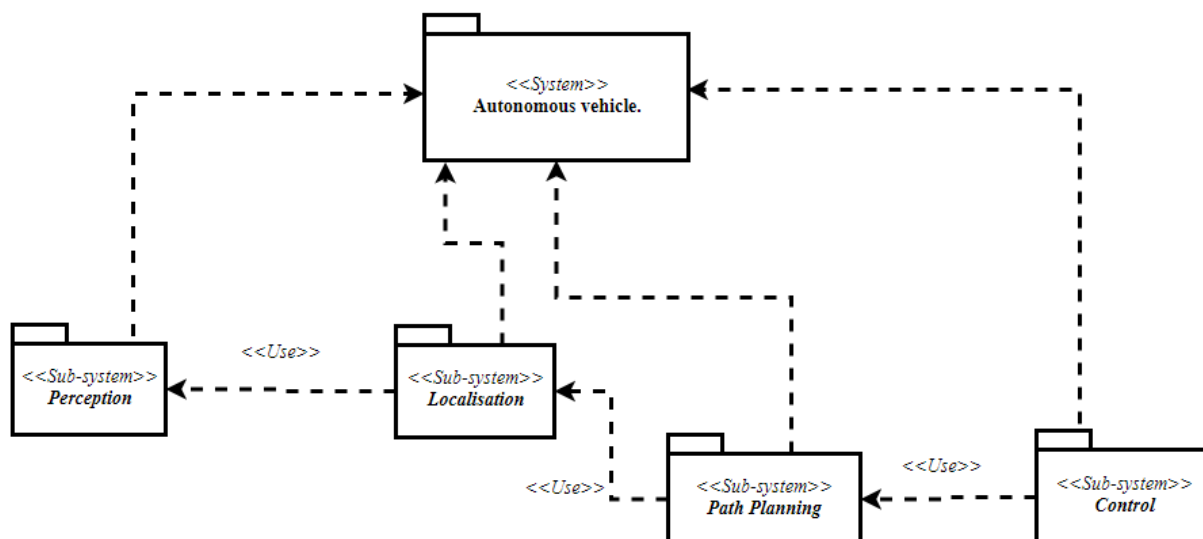


Fig. 2. Components diagram

Рис. 2. Схема компонентов

### Visualisation

LIDAR refers to a light detection and ranging device, which sends millions of light pulses per sec in a well-designed pattern. With its rotational axis, it is able to create a dynamic, three-dimensional map of the environment. LIDAR is currently the heart for object detection for most of the existing autonomous vehicles. Perception en-

tails vision of the system. This is how the computer recognises the environment as we perceive it. In reference we can call in computer vision. It started in the 1950s. This saw the development special techniques such as Canny-edge detection which allows to distinguish the evolution of the colour in an image as shown in Fig. 3.



Fig. 3. Canny-edge detection on a road

Рис. 3. Детектор Кэнни границ на дороге

Perception provides autonomous vehicles with crucial information on the driving environment, including the free drivable areas and surrounding obstacles' locations, velocities, and even predictions of their future states [7]. Based on the sensors used, the environment perception task can be done by using LIDARs, cameras, and/or a fusion between these two kinds of devices. Some other traditional approaches may also involve the use of long/short-range radars and ultrasonic sensors. Perception solves problems like steering Angle Computation, Lane Line Detection and Obstacle & Road Signs/Lights Detection [8]. This subsystem is the first to be developed because all the other subsystems build on from what

was developed for perception. Perception encompasses methods used to be able to visualise data. The given environment is a game. Therefore the developed system should be able to understand the environment. Using computer vision is the fastest way of doing this. Perception entails vision of the system. This is how the computer recognises the environment as we perceive it. In reference we can call in Computer vision. The subsystem will be receiving a constant stream of images. All processes will be happening in real time. Fig. 4 will show how to denoise the image using what is called the canny edge detection. Fig. 5 will show the finding of lane lines after removing everything else [9].



Fig. 4. Canny edge detection of image  
Рис. 4. Четкое распознавание краев изображения



Fig. 5. Finding lanes  
Рис. 5. Поиск полосы движения

Using mathematical models we are able to generate the lane line equation coefficients. First, second, or third-order coefficient equations can be used to approximate lane lines. First-order equations would simply be  $ax+b$  (a straight line) while higher-dimensional ones will allow for curves. Fig. 6 shows this phenomenon.

$$L_s = a x + b, \quad (1)$$

$$L_c = a x^2 + b^2 + c, \quad (2)$$

where  $L_s$  represents a straight line, and  $L_c$  a curve line.



Fig. 6. Lane detection calculation  
Рис. 6. Расчет обнаружения полосы движения

Datasets do not always mention lane line coefficients, and it is also necessary to detect the type of line it is. The lanes can be dashed, solid, double solid or double lanes, one dashed and one solid. It is also important to know whether the line belongs to the ego vehicle lane or to an adjacent one. A single neural network may be really hard to train and harder to generalize if there are multiple features required. Segmentation is a popular approach for solving this problem. The goal is to give a class

to each pixel of a photo. With this approach, each lane corresponds to a class. The neural network aims to generate an image with these colours only. In this type of architecture, the neural network is working in two parts; one - learns the features; Two - part learns the output. Just like for bounding box detection. The module sets the parameters that will be used for the next subsystems. Data is input in the form of video or a stream of images. After performing some mathematical evaluations the

desired result is output. That is the lane lines. The inputs will be images per frame. The input images will have all the noise removed. Then we identify the edges. After that we mask the area we want to focus on. That is when we can detect lane lines. After that we draw the lane lines and they will be the output on the inputted frame.

### Localisation

The following are the different methods used for localisation in autonomous vehicles.

a) Odometry — It uses a starting position and a wheel displacement calculation to estimate a position at a time  $t$ . This technique is generally inaccurate, leading to an accumulation of errors due to measurement inaccuracies, wheel slip.

b) Kalman filter — defines the state of a vehicle.

c) Particle Filter — are a variant of the Bayesian filters. What it does is that it compares the observations of our sensors with the environmental map [10]. This allows us to create particles around areas where the observations are similar to the map.

d) SLAM — Simultaneous Localization And Mapping. This technique uses the position of landmarks to estimate the vehicle's position. The input data will be masked images containing the map. These will have been output from the "Perception Subsystem". Output data will square line on the map and realising where the vehicle is located. This is the actualisation of exactly where the vehicle is positioned in the environment. It is also the implementation of algorithms to estimate where the vehicle is with an error of approximately 10 cm. In this solution, since our environment is a gaming world, there is no GPS system as we might know it. However, there will be a map provided. Games like Asphalt, Need for Speed, GTA and Euro Trucks possess a map on the bottom left corner. The system perception is set for this. The input imaged will have been masked for the map area that is needed in this subsystem. Using computer vision, the system will look where the vehicle is located in the map.

### Path Planning

Path planning is about implementing the brain of an autonomous vehicle. In the gaming world, the map and the road lane will be used for path planning. This process involves Prediction, Decision making and Trajectory generation. Prediction is about trying to make an informed decision of what the objects in the given environment will do in the next few seconds. For handling Prediction, there are 2 main approaches, The Model Based approach and Data Driven approach. In the Model Based approach. All possible trajectories for situation such as highway insertion or intersection, a

model can be drawn up. Options for the vehicle entering the highway are: 1) the vehicle will keep its original lane by speeding up, slowing down or staying at a constant speed; 2) The vehicle may change lanes. Data Driven (machine learning) approach/method is very different. Two stages are defined, a training phase and a prediction phase. The training phase requires collecting large data on the history of vehicles and learning from this data. Unsupervised learning is preferable with this type of data. We use clustering algorithms to define vehicle trajectories. The Frenet coordinates contain two axes, an  $s$  axis indicating the advance relative to the track and a  $d$  axis indicating the distance to the centre of the lane. This marker is the base to estimate if the trajectory deviates from the centre of the lane. Fig. 7 show a comparison of Cartesian plan against the Frenet plane.

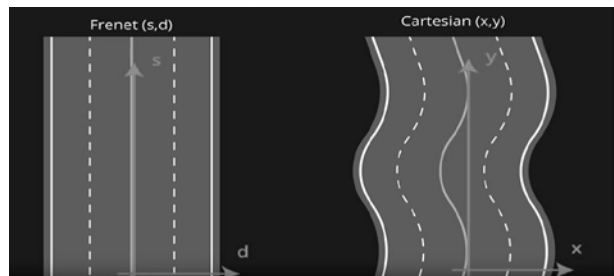


Fig. 7. The Frenet vs the Cartesian plane  
Рис. 7. Френет против декартовой плоскости

Generation of a trajectory requires passing through waypoints created by a fifth level polynomial. The waypoints contain 3 dimensions known as  $S$ ,  $D$ , and  $T$ . These are longitude distance, lateral distance and time passed at any of the points respectively. A curve that goes through all these points is called a trajectory. These points are positioned in space and time. They tell when to move to a specific  $(x, y)$  position and how fast. When taking a decision to overtake a vehicle, the algorithm generates several trajectories for the required decision. It chooses the best trajectory according to the criteria of feasibility, safety, legality, efficiency and comfort. In the case of overtaking for example, the algorithm chooses the best option on the given choices. In an environment called non-holonomic, we are able to make control calculations. In this environment, for instance, the vehicle wheels cannot turn  $90^\circ$ , they are actually in between  $-30^\circ$  and  $30^\circ$ . Taking this into account allows for a more realistic trajectory. Acceleration can be defined between  $-1$  as braking and  $1$  as maximum achievable acceleration. There are 2 implementable models, Kinematic and Dynamic. A kinematic model means that our vehicle realizes

the implementation of mathematical formulas to define the movement and trajectory of the vehicle. The Dynamic model considers the fundamental principles of dynamics. Thus the forces applied to the vehicle like air resistance, vehicle weight, gravity, the contact of the wheels on the ground.

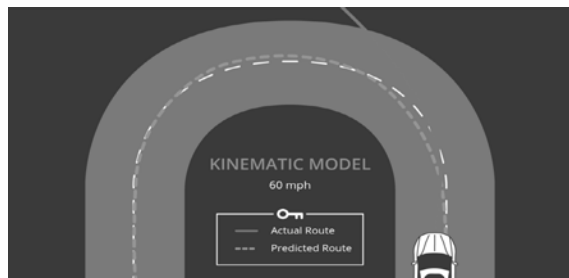


Fig. 8. Kinematic motion model of a vehicle  
Рис. 8. Кинематическая модель движения транспортного средства

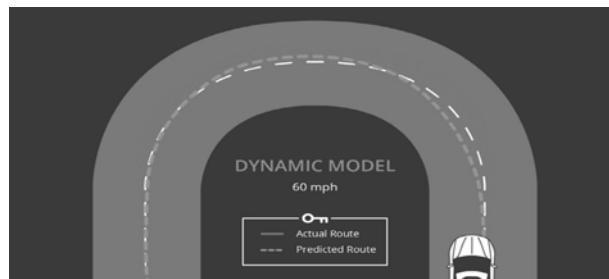


Fig. 9. Dynamic motion model of a vehicle  
Рис. 9. Динамическая модель движения транспортного средства

The input data will be the map screen section. The output data will be the map section with outlined planned trajectory. After getting an estimate of the future prediction, a decision can be made. To make a decision, several criteria is used. The most common are safety, feasibility, efficiency and legality. Finite state machines are a very popular method known for decision making. According to the available situations, all possible states of the vehicle are to be defined. Using cost functions, a choice of the machine state can be made. For each possible scenario, we calculate independent costs and add them up. The lowest cost scenario wins. To calculate a dangerous manoeuvre:

$$C_T = 5C_F + 4C_S + 3C_L + 2C_C + C_{Speed}, \quad (3)$$

where  $C_T$  is the total cost,  $C_F$  is the feasibility cost,  $C_S$  is the security cost,  $C_L$  is the legality cost,  $C_C$  is the comfort cost,  $C_{Speed}$  is the speed cost.

We make use of a different coordinate system than the Cartesian coordinate system. This is because the Cartesian coordinate system takes into account the dimension (x, y) but does not make sense if we want to find one's bearings in relation to the road. This is where the trajectory is generated.

### Control

To make vehicles autonomous, the control stage is very important and crucial. Control relies on all the previous modules: perception, localization and planning, which must be developed perfectly. Controllers are unique to each vehicle which makes it difficult to export them. An autonomous vehicle uses the perception module to

The centrifugal is not taken into account in a Kinematic model. It is a force that can cause a vehicle to leave its trajectory in a bend. Fig. 8 and Fig. 9 show the Kinetic and Dynamic models respectively.

grasp its environment, the localization module to understand its position during this environment and also the planning module to form decisions and generate trajectories. The control module is now answerable for moving the vehicle by generating an angle for the hand wheel and an acceleration. Taking into consideration the road and therefore the vehicle, and integrating physics, we develop algorithms to follow the waypoints efficiently. This moves the vehicle by operating the brake, acceleration and wheel. An impact algorithm is named a controller. The input data will be a combination of the lane lines and the map trajectories. The output data will be keyboard controls relating to the projected trajectory and lane lines. As a controller a Proportional Integral Derivative PID controller was used [11]. It is an algorithm that calculates values like a steering wheel angle from an error calculation. The difference between the intended trajectory and the one adopted. It is the simplest and most common controller at the moment. It can be implemented quickly and can operate in simple situations. In the case of stand-alone vehicles, a PID controller can calculate the angle of steering and another to calculate the acceleration. It is impossible to model the physics of the vehicle. When driving, it is natural to adjust manoeuvre according to the size, mass and dynamics of the vehicle. A PID controller cannot do this. The PID controller has proportional, integral and derivative elements as in its name [12]. Fig. 10 shows how PID controllers work.

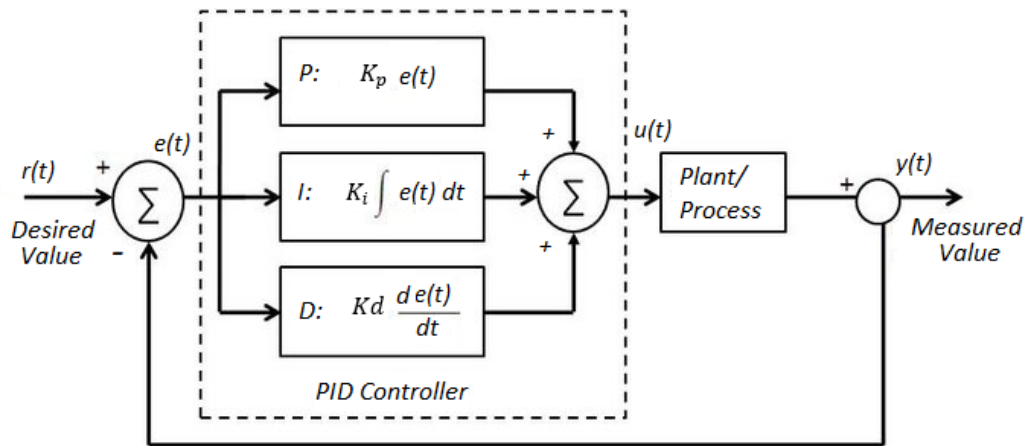


Fig. 10. The PID controller  
Рис. 10. ПИД-регулятор

Proportional refers to a correction to the steering wheel proportional to the error. If too far from the goal, turn the wheel in the other direction. The main disadvantage of a single P controller is that it causes a constant oscillation. This oscillation is more or less important depending on the frequency at which the algorithm calculates the error. The coefficient  $K_p$  indicates the degree of the desired oscillation. The proportional element formula is depicted in equation 4:

$$P = K_p e(t), \quad (4)$$

Derivative - this for suppressing the oscillation effect by adding a term that dampens the formula. It is the change of error. It does this by decreasing and reducing slightly the angle it adopts when approaching a smooth path. Fig. 13 shows how the D controller is represented on a graph:

$$D = K_d \frac{de(t)}{dt}, \quad (5)$$

Integral - is used to correct the mechanical error that causes the wheel to turn a lot depending on the vehicle's stability. This is done by adding a last term to the sum of cumulative errors. Therefore, it is a sum of 3 components that allow the vehicle to follow effectively, a trajectory in real time. To optimize driving,  $K_p$ ,  $K_i$ ,  $K_d$  coefficients are required. Figure 5.4 will show the combination of all these controllers and the optimum PID we need.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}. \quad (6)$$

The controller gets instructions for the vehicle like the wheel angle or acceleration level taking into consideration the particular constraints and also the trajectory generated. In the world of robot-

ics, there are a large number of controllers, depending on the type of robot needed to be moved. The environment can also play a role.

### System algorithms

Figures 11–15 show the algorithms for each subsystem. Figure 11 is the algorithm of the whole system. From Fig. 12 to Fig. 15, these are the algorithms of each subsystem.

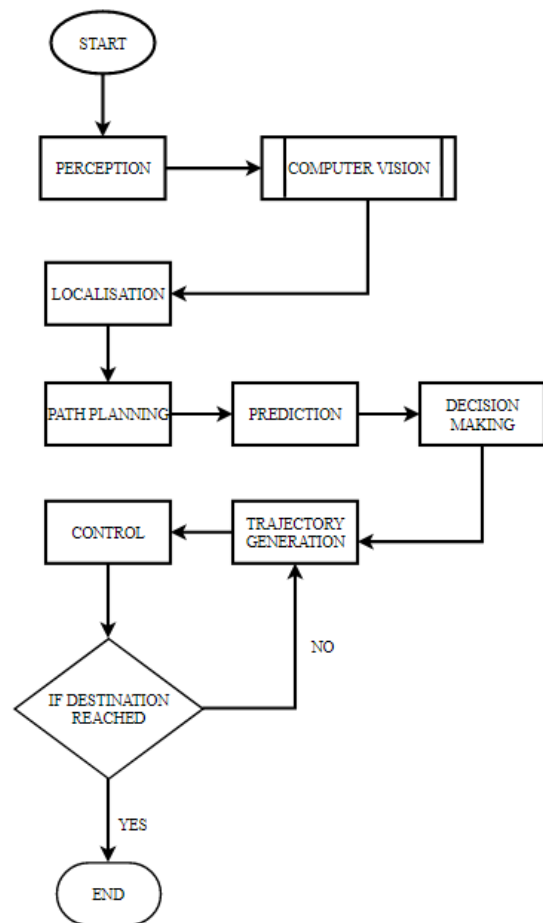


Fig. 11. Diagram of how system works  
Рис. 11. Схема работы системы

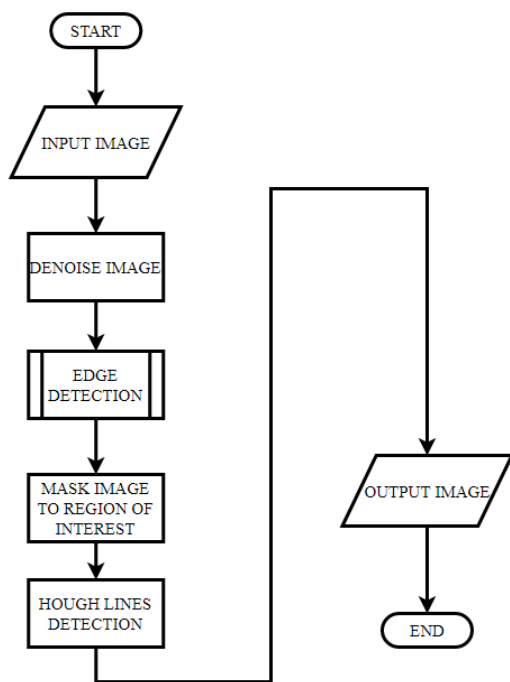


Fig. 12. Lane line detection algorithm  
 Рис. 12. Алгоритм определения полосы движения

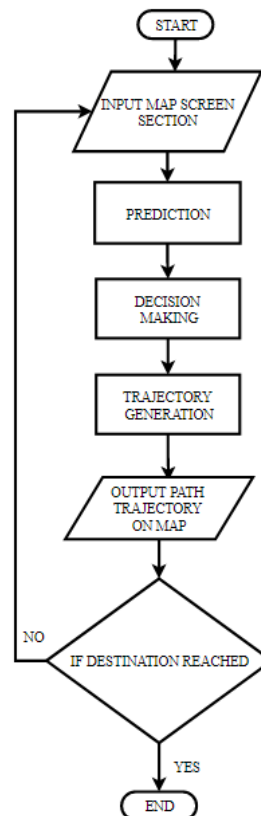


Fig. 14. Path planning algorithm  
 Рис. 14. Алгоритм планирования пути

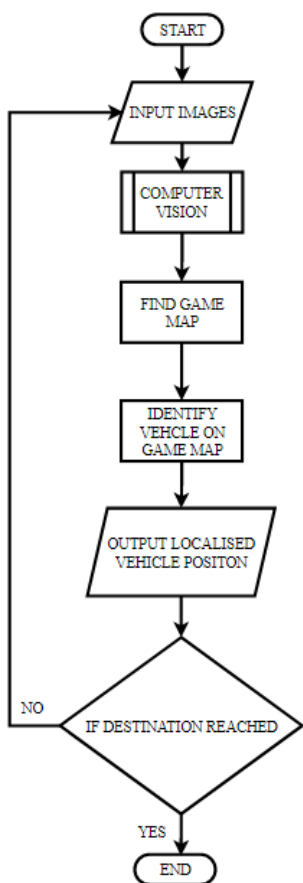


Fig. 13. Localisation algorithm  
 Рис. 13. Алгоритм локализации

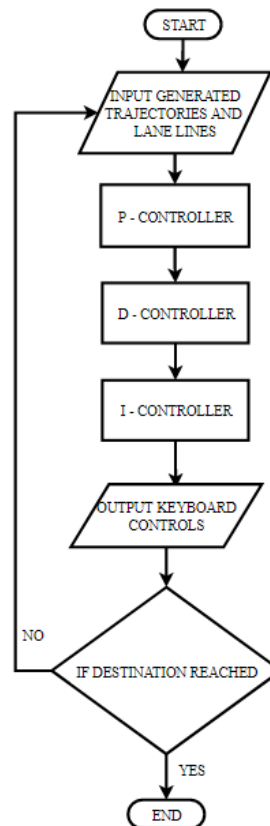


Fig. 15. Control algorithm  
 Рис. 15. Алгоритм управления



### Experiments and results

Each subsystem had an experiment conducted, therefore five experiments were carried out. For visualisation, we were trying to achieve computer perception [13]. If the output is simply a black image with colours, it is then very simple to use machine learning and linear regression to find the lane lines on the same colour points [14]. The input data was the video of the gaming screen available, e.g., Need for Speed, or The Run. The output information will be: a) Road lane lines detected, b) Map detected. Figures 16-18 show 3 different camera views of lane detection. Fig. 19 shows the second result of the experiment. It shows the lanes as well as the map identified. This is necessary in order to have inputs for the localisation module.



Fig. 16. Camera view 1  
Рис. 16. Вид камеры 1



Fig. 17. Camera view 2  
Рис. 17. Вид камеры 2



Fig. 18. Camera view 3  
Рис. 18. Вид камеры 3

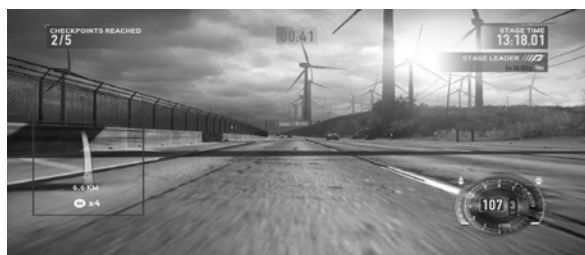


Fig. 19. Map recognized  
Рис. 19. Определение карты

After finding the map, the only step left to achieve localisation is pin pointing the exact location of the vehicle. Fig. 20 shows the actualisation of exactly where the vehicle is positioned in the environment.



Fig. 20. Localised Vehicle  
Рис. 20. Локализация автомобиля

Having localized the vehicle, the system had to plan the path of the vehicle. Fig. 21 and Fig. 22 will show the map section with outlined planned trajectory.



Fig. 21. Camera view  
Рис. 21. Вид камеры



Fig. 22. Map view showing trajectory generated  
Рис. 22. Карта, показывающая сгенерированную траекторию

### Conclusion

As a result of the work, the analysis of the subject area was carried out. In the process of completing the final qualifying work the set goals and objectives were fully met. The system comprises of four subsystems. All four subsystems were completed. The subsystems are development of the perception subsystem, the localisation subsystem, the path planning subsystem and the control subsystem. In particular, there was developed a software product that simplifies automation but only limited to gaming environ-

ments. The main advantage is that the user does not have to develop a game to carry out the automation system experiments. The system is free and cheaper and the only required is a computer capable of playing a computer game. Because the system is free, a lot of researchers can use it and work on autonomous projects and improve on them before acquiring expensive hardware.

### References

1. Feng, L., Everett, H. R., Borenstein, J. (1994). "Where am I?": sensors and methods for autonomous mobile robot positioning. University of Michigan.
2. Katrakazas, C., Qudus, M., Chen, W.-H., & Deka, L. (2015). *Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions*. Transportation Research Part C: Emerging Technologies, 60, 416–442. doi.org/10.1016/j.trc.2015.09.011.
3. Pendleton, S., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y., Rus, D., & Ang, M. (2017). *Perception, Planning, Control, and Coordination for Autonomous Vehicles*. Machines, 5(1), 6. Available at: <https://doi.org/10.3390/machines5010006>.
4. Badue, C., et al. (2019). *Self-Driving Cars: A Survey*. arXiv:1901.04407v2.
5. Minton, S. (1993). *The Morgan KAUFMANN Series in Machine Learning*. Machine Learning Methods for Planning, p. ii. doi:10.1016/b978-1-4832-0774-2.50001-9.
6. Ganguly, S. (2021). *Machine Learning. Quantum Machine Learning: An Applied Approach*. Apress. pp. 41–97.
7. Brownlee, J. (2020). *10 Clustering Algorithms with Python. Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/clustering-algorithms-with-python>.
8. Cohen, J. (2021). *Deep Learning Algorithms in Self-Driving Cars*. Medium. Available at: <https://becominghuman.ai/deep-learning-algorithms-in-self-driving-cars-14b13a895068>.
9. *Road Lane line detection-Computer Vision Project in Python*. DataFlair. (2021). Available at: <https://data-flair.training/blogs/road-lane-line-detection>.
10. Ferdinand, N. (2020). *A Deep Dive into Lane Detection with Hough Transform*. Medium. Available at: <https://towardsdatascience.com/a-deep-dive-into-lane-detection-with-hough-transform-8f90fdd1322f>.
11. Johnson M. (2005). PID Control Technology. In: Johnson M.A., Moradi M.H. (eds) PID Control. Springer, London. Available at: [https://doi.org/10.1007/1-84628-148-2\\_1](https://doi.org/10.1007/1-84628-148-2_1).
12. Crowe J., Johnson M. (2005). Phase-Locked Loop Methods and PID Control. In: Johnson M.A., Moradi M.H. (eds) PID Control. Springer, London. Available at: [https://doi.org/10.1007/1-84628-148-2\\_7](https://doi.org/10.1007/1-84628-148-2_7).
13. Semmlow J. L. (2008). *Fundamentals of Image Processing: Matlab Image Processing Toolbox*. Bio-signal and Medical Image Processing, pp. 293–325.
14. Kittler, J. (1985). *Image Processing System Architectures*. Research Studies Press, Wiley.

\*\*\*

### Применение машинного обучения для визуализации и планирования движения автономных транспортных средств

Рутендо Медзи, ИЖГТУ имени М. Т. Калашникова, Ижевск, Россия  
М. А. Аль Аккад, ИЖГТУ имени М. Т. Калашникова, Ижевск, Россия

Данная статья посвящена описанию системы для управления транспортными средствами в игровой среде, при этом целью работы является моделирование и экспериментальное исследование работы беспилотных транспортных средств. Долгосрочной целью данного исследования является исключение человеческого фактора. Разработаны подсистемы восприятия, локализации, планирования и управления. В дополнение к обычной веб-камере в работе были использованы датчики LiDAR и RADAR. После получения информации от модуля восприятия система локализует позицию транспортного средства, далее для составления оптимального пути используется модуль планирования, который определяет, куда будет перемещаться транспортное средство, используя данные модуля локализации. После определения оптимального пути система осуществляет управление транспортным средством, таким образом, чтобы оно двигалось автономно, без помощи человека. В качестве регулятора использовался пропорционально-интегрально-производный ПИД-регулятор. При разработке системы использовались: язык программирования Python, компьютерное зрение и машинное обучение. Для обеспечения работы системы необходим компьютер с графическим процессором и мощной графической картой, которая может запускать игру с автомобилем, дорогами с полосами движения и картой дороги. При соответствующей доработке, разработанная система может стать хорошим инструментом при проведении экспериментов по достижению надежной автономной навигации транспортного средства, также может быть использована в качестве базы для работы над автономными транспортными средствами с реальными аппаратными устройствами.

**Ключевые слова:** автономные транспортные средства, восприятие, управление, локализация, навигация, планирование движения.

Получено: 11.08.2021