PATH PLANNING FOR AUTONOMOUS VEHICLES

(AKILLI ARAÇLAR İÇİN YÖRÜNGE PLANLAMA)

by

AHMET EMRE DANIŞMAN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

SEP 2019

Approval of the thesis:

PATH PLANNING FOR AUTONOMOUS VEHICLES

submitted by **AHMET EMRE DANIŞMAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Galatasaray University** by,

Examining Committee Members:

Prof. Dr. Tankut Acarman
Supervisor, **Computer Engineering Department, GSU** ____________

Prof. Dr. Mehmet Turan Söylemez
Control and Automation Engineering, İTÜ ____________

Assoc. Prof. Dr. Murat Akın
Computer Engineering Department, GSU ____________

Date: ____________

ACKNOWLEDGMENTS

First of all, I would like to express my very great appreciation to my supervisor, Prof. Dr. Tankut Acarman for his expert advice and guidance.

I am grateful to Çağdaş Yaman for his great contribution and help.

My special thanks are to my family for their continuous support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Scientific progress is an ongoing process and humans always have new ideas for new inventions. Although, scientific process is cumulative, you must invent the tire before inventing the car. Currently, mankind have past that point and looking towards to new challenges. One of the new ideas is cars that drive themselves, in a more formal and general way, the 'autonomous ground vehicles'. Autonomous vehicles in city traffic have been a dream for a long time. However, this great idea comes with its own set of problems. There are many parts of autonomous driving such as scene understanding, path planning. Many methods have been studied and developed to solve its problems. In this thesis, a novel path planning algorithm and results of extensive experimentation on it are presented. The algorithm runs real-time and is an alternative to well known existing algorithms. The experiments are done in a scalable custom environment consisting of a custom-built vehicle, desktop computer, a camera and an open source marker library.

Keywords : Autonomous ground vehicles, path planning, real-time computing, robotics

ÖZET

Bilimsel ilerleme sürekli devam eden bir süreçtir ve insanların her zaman yeni icatlar için yeni fikirleri vardır. Ancak, bilimsel ilerleyiş kümülatiftir, arabayı icat etmeden önce tekerleği icat etmeniz gerekmektedir. An itibarıyla insanlık o noktayı aşmış bulunmaktadır ve yeni meydan okumalara açıktır. Yeni fikirlerden biri kendileri giden arabalar, daha resmi ve genel bir tabirle, 'otonom yer araçları'dır. Şehir trafiğinde seyreden otonom araçlar uzun yıllardan beri bir hayal olmuştur. Ama bu fikrin gerçekleşmesi için bazı zorlukların üstesinden gelinmesi gerekmektedir. Otonom araçların senaryo algılama ve yörünge planlama gibi pek çok ayrı bileşeni vardır. Sorunlarını çözmek için birçok metod üzerine çalışılmış ve geliştirilmiştir.
Bu tezde, yeni bir yörünge planlama algoritması tanıtılmış ve üzerinde yapılan detaylı deneylerin sonuçları paylaşılmıştır. Algoritm gerçek amanlı çalışmaktadır ve bilinen algoritmalara bir alternatiftir. Deneyler özel olarak ayarlanmış ölçeklenebilir bir ortamda, el yapımı bir araç, bilgisayar, kamera ve açık kaynak kodlu bir karekod kütüphanesi kullanılarak yapılmıştır.

Anahtar Kelimeler : Otonom araçlar, yörünge planlama, gerçek zamanlı programlama, robotik

1 INTRODUCTION

One of the fundamental problems of creating an autonomous ground vehicle is the navigation. It can be defined as the set of manoeuvres that enables the vehicle to move from one location to another safely. In order to call this navigation process intelligent, the autonomous ground vehicle should be respecting the environment and its surroundings during it's movement towards the goal. This can be achieved by intelligent algorithms that take the obstacles and the capability of the autonomous ground vehicle into the account. So, it has been a research topic for quiet some time and many different algorithms and approaches have been studied over time [5]. More information on brief history of autonomous ground vehicles and different fundamental algorithms can be found in the following sections.

1.1 History of Autonomous Ground Vehicles

Ever since the first time the idea of autonomous ground vehicle has come to mind of mankind, we had tried to solve the problems it created with using all the available technology and power. One of the problems that have risen is the problem of navigating autonomous ground vehicle, which has proved itself to be a challenging and interesting problem over the years. Throughout the history, mankind approached this problem many times, from many different angles and it has been studied extensively in the field of robotics and intelligent vehicles for decades [6].
In 1925, Houdina Radio Control demonstrated the radio-controlled "American Wonder" through traffic jam. The American Wonder was equipped with a transmitting antenna and was operated by a second car that followed it [7]. Another early representation of an autonomous ground vehicle was Norman Bel Geddes's Futurama exhibit at the 1939 World's Fair, which depicted radio-controlled electric cars that were propelled via electromagnetic fields provided by circuits embedded in the roadway [8].
In 1957, a full size system was successfully demonstrated by RCA Labs on a 400-foot strip of public highway in Nebraska, United States of America. A series of experimental detector circuits buried in the pavement were a series of lights along the edge of the road. The detector circuits were able to send impulses to guide the car and determine the presence and velocity of any metallic vehicle on its surface [9].
In 1980s, the Defense Advanced Research Projects Agency (DARPA) funded Autonomous Land driven Vehicle (ALV) project in the United States made use of new

technologies developed by the University of Maryland, Carnegie Mellon University, the Environmental Research Institute of Michigan, Martin Marietta and SRI International. The ALV project achieved the first road-following demonstration that used lidar, computer vision and autonomous robotic control to direct a robotic vehicle at speeds of up to 19 miles per hour (31 km/h). In 1987, HRL Laboratories (formerly Hughes Research Labs) demonstrated the first off-road map and sensor-based autonomous navigation on the ALV. The vehicle traveled over 2,000 feet (610 m) at 1.9 miles per hour (3.1 km/h) on complex terrain with steep slopes, ravines, large rocks, and vegetation. By 1989, Carnegie Mellon University had pioneered the use of neural networks to steer and otherwise control autonomous vehicles, forming the basis of contemporary control strategies [10].

In 1991, the United States Congress passed the ISTEA Transportation Authorization bill, which instructed USDOT to "demonstrate an automated vehicle and highway system by 1997." The Federal Highway Administration took on this task, first with a series of Precursor Systems Analyses and then by establishing the National Automated Highway System Consortium (NAHSC). This cost-shared project was led by FHWA and General Motors, with Caltrans, Delco, Parsons Brinkerhoff, Bechtel, UC-Berkeley, Carnegie Mellon University, and Lockheed Martin as additional partners. Extensive systems engineering work and research culminated in Demo '97 on I-15 in San Diego, California, in which about 20 automated vehicles, including cars, buses, and trucks. The demonstrations involved close-headway platooning intended to operate in segregated traffic, as well as "free agent" vehicles intended to operate in mixed traffic. While the subsequent aim was to produce a system design to aid commercialization, the program was cancelled in the late 1990s due to tightening research budgets at USDOT. Overall funding for the program was in the range of $90 million [11].

In the first Grand Challenge held in March 2004, DARPA offered a $1 million prize to any team of robotic engineers which could create an autonomous car capable of finishing a 150-mile course in the Mojave Desert. No team was successful in completing the course [12]. In October 2005, the second DARPA Grand Challenge was again held in a desert environment. GPS points were placed and obstacle types were located in advance. This year, five vehicles completed the course and Stanley University vehicle ended up winning the competition. In November 2007, DARPA again sponsored Grand Challenge III, but this time the Challenge was held in an urban environment. In this race, a 2007 Chevy Tahoe autonomous car from Carnegie Mellon University earned the 1st place [13].

Many major automotive manufacturers, including General Motors, Ford, Mercedes Benz, Volkswagen, Audi, Nissan, Toyota, BMW, and Volvo, are testing driverless car systems as of 2013. BMW has been testing driverless systems since around 2005 [14], while in 2010, Audi sent a driverless Audi TTS to the top of Pike's Peak at close to race speeds [15]. In 2011, GM created the EN-V (short for Electric Networked Vehicle), an autonomous electric urban vehicle [16]. In 2012, Volkswagen began testing a "Temporary Auto Pilot" (TAP) system that will allow a car to drive itself at speeds of up to 80 miles per hour (130 km/h) on the highway [17]. Ford has conducted extensive research into driverless systems and vehicular communication systems [18]. In January 2013, Toyota demonstrated a partially self-driving car with numerous sensors and communication systems [19].

The focus of this thesis is on the path planning of self-made mini autonomous ground vehicle, more information on the vehicle specifications is on the following chapters. In the next chapter, we will review the relevant literature in the field of path planning along with simulation results obtained using these techniques. Then, the contribution of this thesis is discussed. Finally, we will provide the outline and organization of this thesis.

2 RELATED WORK

Path planning algorithms can be defined in a very simple way, they find a collision-free path from an initial point to a final point. Simplicity of this definition is the ultimate sophistication that allows so many different approaches for path planning algorithms. Over the decades of collaborative study, there are some fundamental algorithms which are the basis for almost every other algorithm. These algorithms include, but not limited to; probabilistic roadmap, rapidly-exploring random tree and potential field algorithms.

2.1 A*

A* is a heuristic, best-first search algorithm that is based on Dijkstra's Algorithm [20] and works on graphs. So in order to make this algorithm work on a 2-dimensional space, the free space in the environment must be divided to grids as shown in Figure 2.1. The algorithm starts from a given starting node of a graph, or a grid, and aims to find a path to the given goal node having the smallest cost. The cost can be defined as the total distance travelled, total time spent, total cost of the path etc. A* algorithm maintains a tree of paths originating at the start node and extending those paths one edge at a time until its termination criterion is satisfied or it reaches the destination node of the graph. This extension makes the algorithm potentially able to search a huge area of the map. Based on the scenario, it may lead to optimal or near-optimal solutions [1].

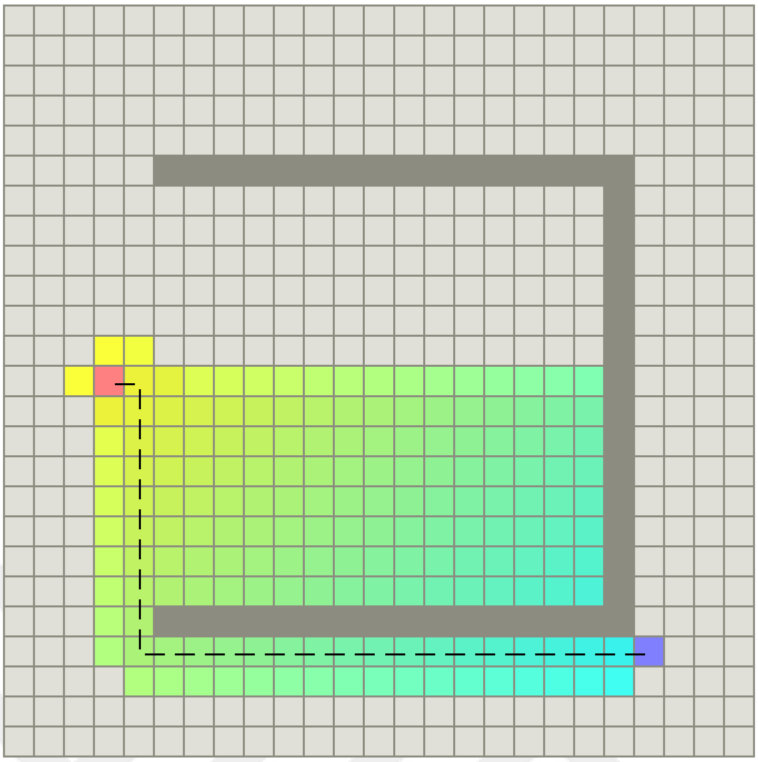

Figure 2.1: A* Algorithm [1]

2.2 Probabilistic Roadmap

Probabilistic Roadmap algorithm is a geometry based planner that works on multi-dimensional spaces, but more practical in fewer dimensional spaces. It takes random sample points from the configuration space, tests them for whether they are in the free space or not then and uses a local planner to attempt to connect these configurations to other nearby configurations. The starting and goal configurations are added in and a graph search algorithm is applied to the resulting graph to determine a path between the starting and goal configurations [2].

In Figure 2.2, you see an example roadmap generated by the algorithm. The gray areas represent obstacles, circles represent the nodes of the roadmap, lines between circles represent the edges of the roadmap the thick lines represent the generated path from the initial point to the goal point. In this particular example, the number of k closest neighbours for the construction is set to three. Randomly selected points in space are connected to three closest nodes, if their distance less than the threshold.

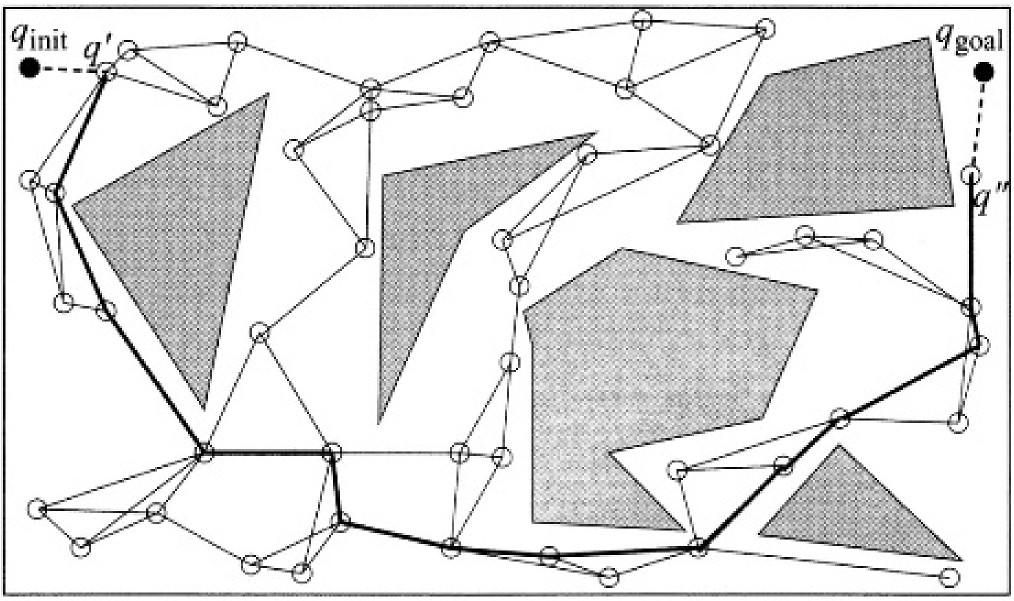

Figure 2.2: Probabilistic Roadmap Algorithm [2]

2.3 Rapidly-Exploring Random Tree

Rapidly-Exploring Random Tree (RRT) is an efficient algorithm that works on multi-dimensional spaces by randomly building a space-filling tree. The tree is constructed by randomly selected from the free space [3]. RRTs grow a tree rooted at the initial point by adding random points to the nearest point in the tree, given that the randomly selected point does not break any constraints. In Figure 2.2, blue circles represent obstacles, black lines represent the connection between the points and the red line represents the resulting path from the initial point to the goal point.

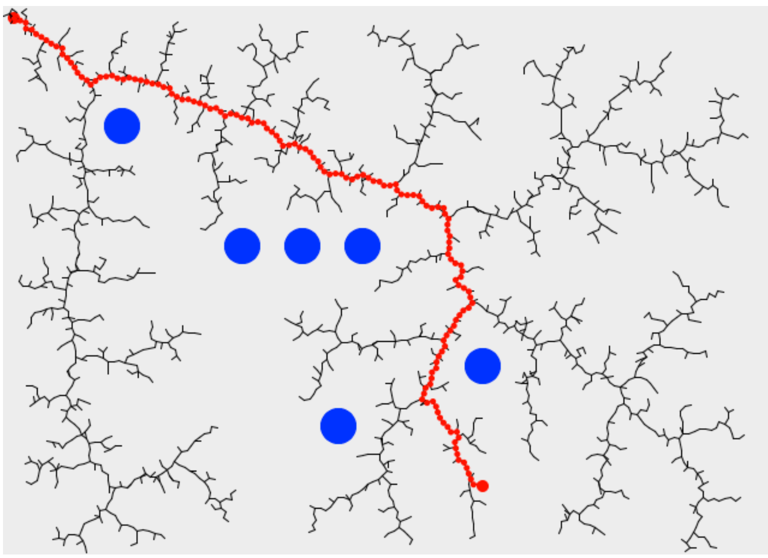

Figure 2.3: Rapidly-Exploring Random Tree [3]

2.4 Potential Field

Potential field algorithms were inspired from the concept of electrical charges. Imagine the vehicle as a electrically-charged particle, then obstacles in the space should have the same type of electrical charge in order to repulse the vehicle and avoid any collision. For the same reason, the goal point should have the opposite type of electrical charge that will attract the vehicle. In Figure 2.4, red circles represent obstacles, blue point represents the initial point, arrows represent the direction and amount of force applied to the vehicle and green line represents generated path. [4]

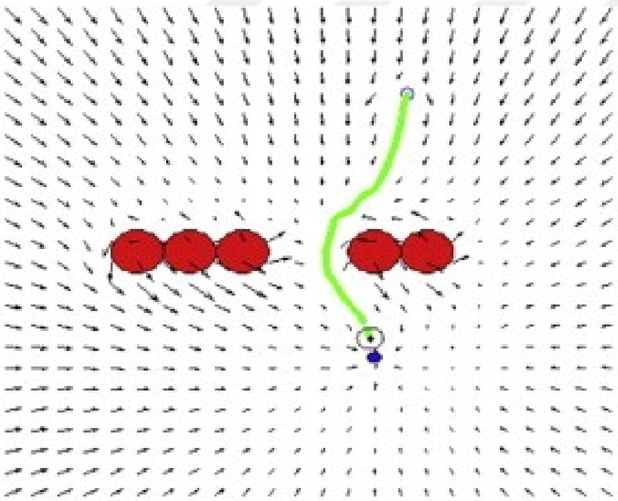

Figure 2.4: Potential Field [4]

3 METHODOLOGY

The algorithm proposed in this paper works real-time and is a novel algorithm that can be alternative to the algorithms mentioned in the previous chapter. Our proposed algorithm is inspired from them, and improves them in certain ways. In this chapter, our novel algorithm will be described.

The algorithm can be explained in 2 parts for the sake of simplicity. The first part of the algorithm is based on a derived version of Hybrid A*[21] and the second step is curve-fitting. Hybrid A* algorithm is an improved version of the well-known A* algorithm, it also takes a basic vehicle model into account while creating path in addition to basic functionality of A*. It does not generate any extremely sharp turns and the generated path is smoother in general. The first part of our algorithm adds up on Hybrid A* and makes improvements on it by smoothening it even further and directing it towards some local goal points. The goal points are deduced from the obstacles and the final destination point.

The second part of our algorithm treats each obstacle in the experimentation space that blocks the way separately. It fits curves around them to avoid collision and connects them with the path parts that the first part generates. Eventually, the path is produced respecting the obstacles.

After the path is finalized, we direct the vehicle to follow it and make it possible with a custom controller. The controller controls the steering and torque. It calculates the steering using two parameters. It adjusts the steering value given by the path depending on these parameters. One parameter is the difference between the current steering value and the steering value that should the car have at that moment. The other parameter is the offset, the distance that vehicle has drifted away from the given path. After calculating the parameters, our controller calculates the new steering value and sends it to vehicle. The controller controls the torque to command the vehicle to go forward or backwards. Trajectory planner is also able to output paths requiring the vehicle to drive backwards and the vehicle is capable of doing it.

4 EXPERIMENTAL STUDY

In order to analyze the performance of our algorithm, we have done numerous experiments with our custom setup.The main components of the setup are the vehicle, the desktop PC and the camera. Details of hardware components of our experimentation setup are given in this chapter. The focus of the experimentation was testing the ability of feasible path generation and following the generated path as well as re-routing if needed. The main parameters of our experiment are vehicle's starting position, starting direction and obstacles on the field.

4.1 Experimental Setup

We have built the vehicle from scratch using the LEGO Mindstorms NXT 2.0 set [22]. The vehicle is a 4-tire car sized 24 cm long, 17 cm wide 14 cm high. It has 3 electric motors on it, 2 of which powers rear-wheel driving and 1 control the steering. The motors are connected to the 'intelligent brick' that has a 32-bit ARM7 microprocessor [23]. The intelligent brick is connected to the desktop PC via Bluetooth.

The steering on the car applies Ackermann steering geometry, to better simulate the real life autonomous cars. Ackermann steering geometry allows the wheels on the inside and outside of a turn to trace out circles of different radii by turning them for different angles, inner tire turning more [24].

On top of the vehicle and on the ground, as obstacles, we have ArUco markers [25]. Markers are used to detect the vehicle's and obstacles' positions and orientations.

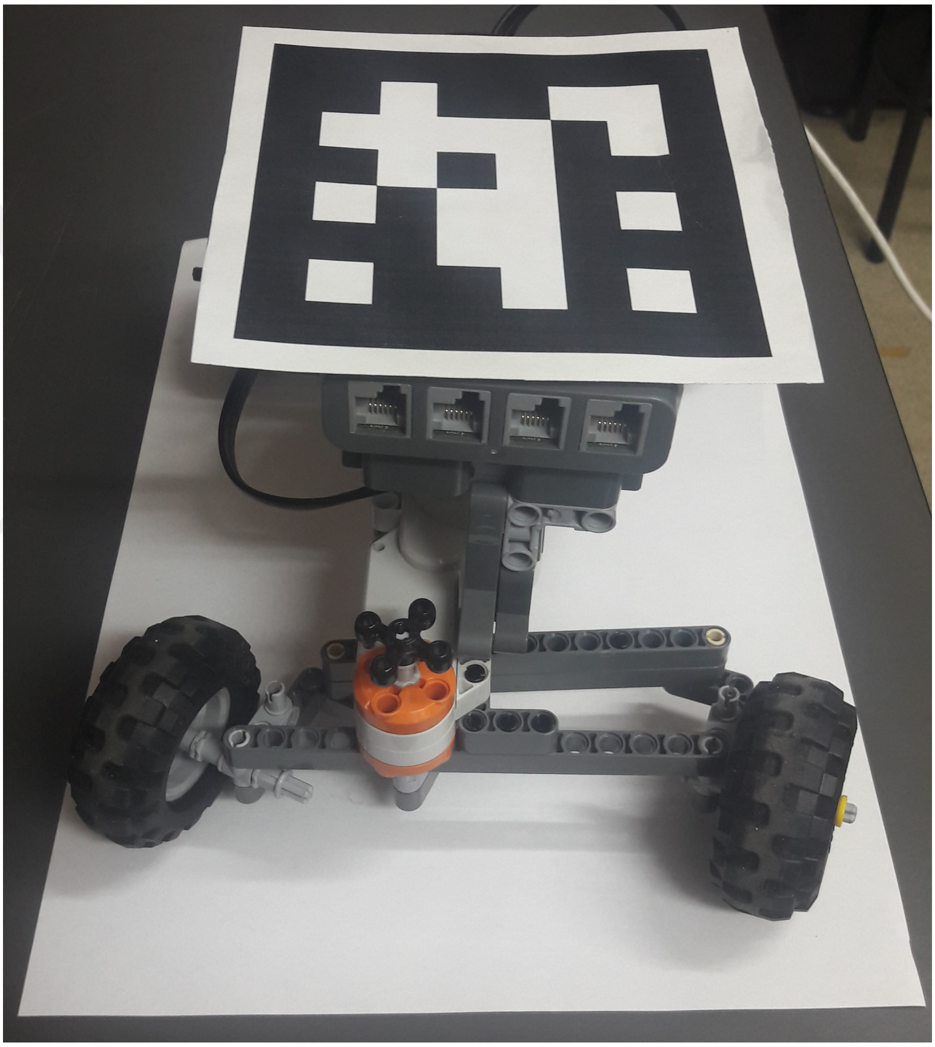

Figure 4.1: Front View of the Vehicle

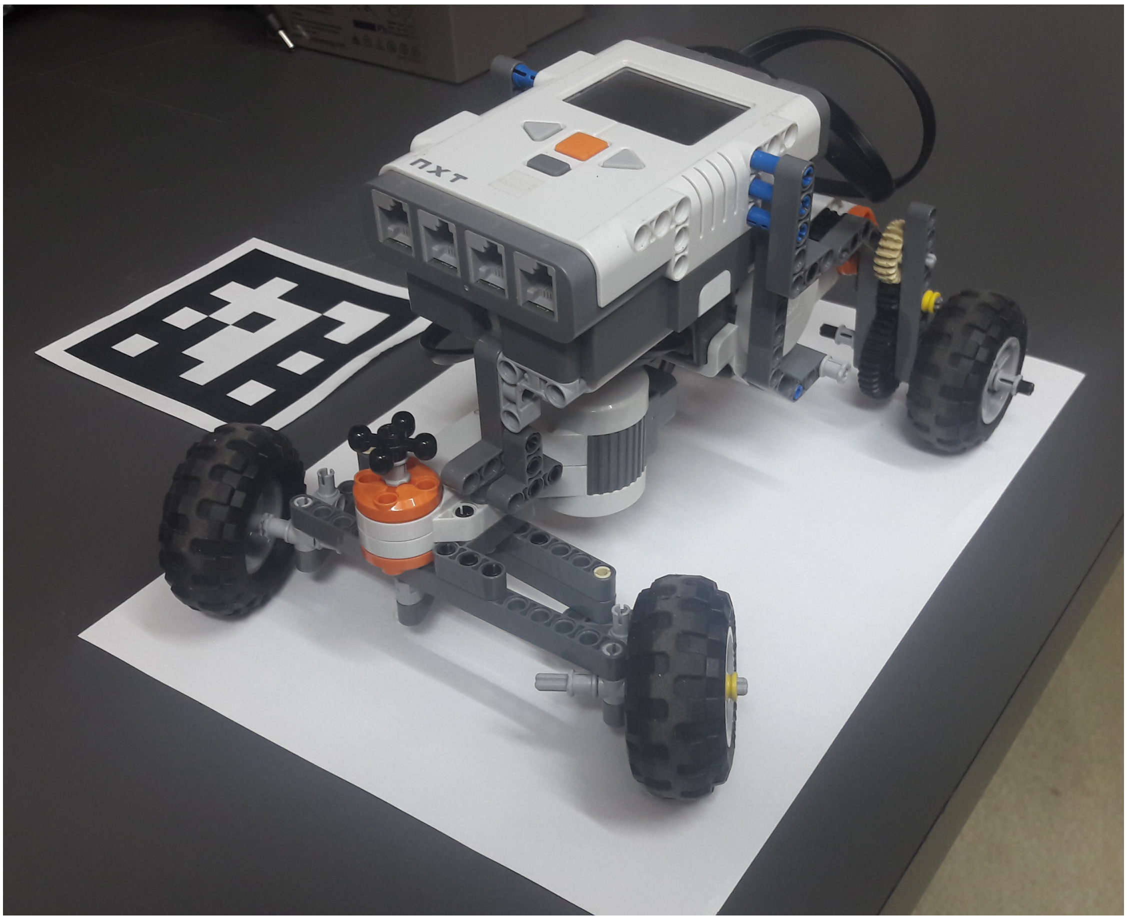

Figure 4.2: Vehicle and ArUco Marker

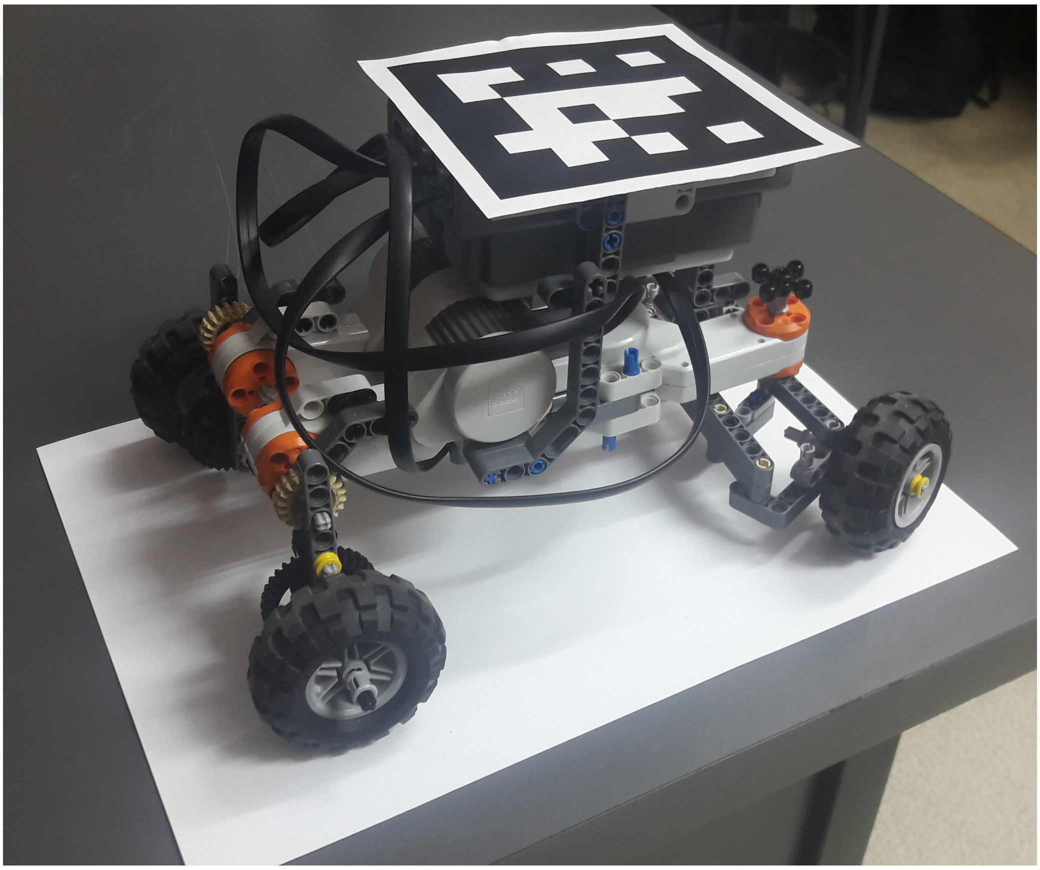

Figure 4.3: Dimetric View of the Vehicle

We have used StereoLab's ZED 2k Stereo Camera as the visual sensor. The camera allows us to get high-resolution image, so that we can accurately calculate the position and orientation of the components and plan out path accordingly. Also, the camera has a high Frame Per Second (FPS) value that enables us to check the vehicle's position as often as possible and take actions as quickly as possible, if any needed.
The ZED camera has wide-angle all-glass dual lens with reduced distortion. It's field of view is 90° (H) x 60° (V) x 110° (D) max. It can capture 1080p, 3840 x 1080, videos at 30 FPS. It uses real-time depth-based visual odometry and SLAM for moving recording, has 6-axis pose accuracy up to 100 Hz. The camera is connected to the desktop PC via a USB 3.0 port [26].

Table 4.1: Camera Specs

Video Mode	1080p
Frames Per Second (FPS)	30
Output Resolution (side by side)	3840x1080
Depth Resolution	3840x1080
Depth Format	32-bit
Depth Range	0.5 - 20 m
Stereo Baseline	120 mm

The desktop PC is also custom built, components chosen to maximize computing power. The processor is Intel Core i7-8700k which has 6 cores, 12 threads, 12MB SmartCache and runs at 3.70GHz. The graphics card is NVIDIA GeForce GTX1080Ti with 11GB RAM, 3584 CUDA cores and 352-bit memory interface width. The motherboard is MSI Z30M Gaming Pro AC with 32GB of RAM on it.

Table 4.2: CPU Specs

Lithography	14 nm
Number of Cores	6
Number of Threads	12
Processor Base Frequency	3.70 GHz
Cache	12 MB SmartCache
Bus Speed	8 GT/s DMI3

Table 4.3: GPU Specs

NVIDIA CUDA® Cores	3584
Memory Speed	11 Gbps
Standard Memory Config	11 GB GDDR5X
Memory Interface Width	352-bit
Memory Bandwidth (GB/sec)	484

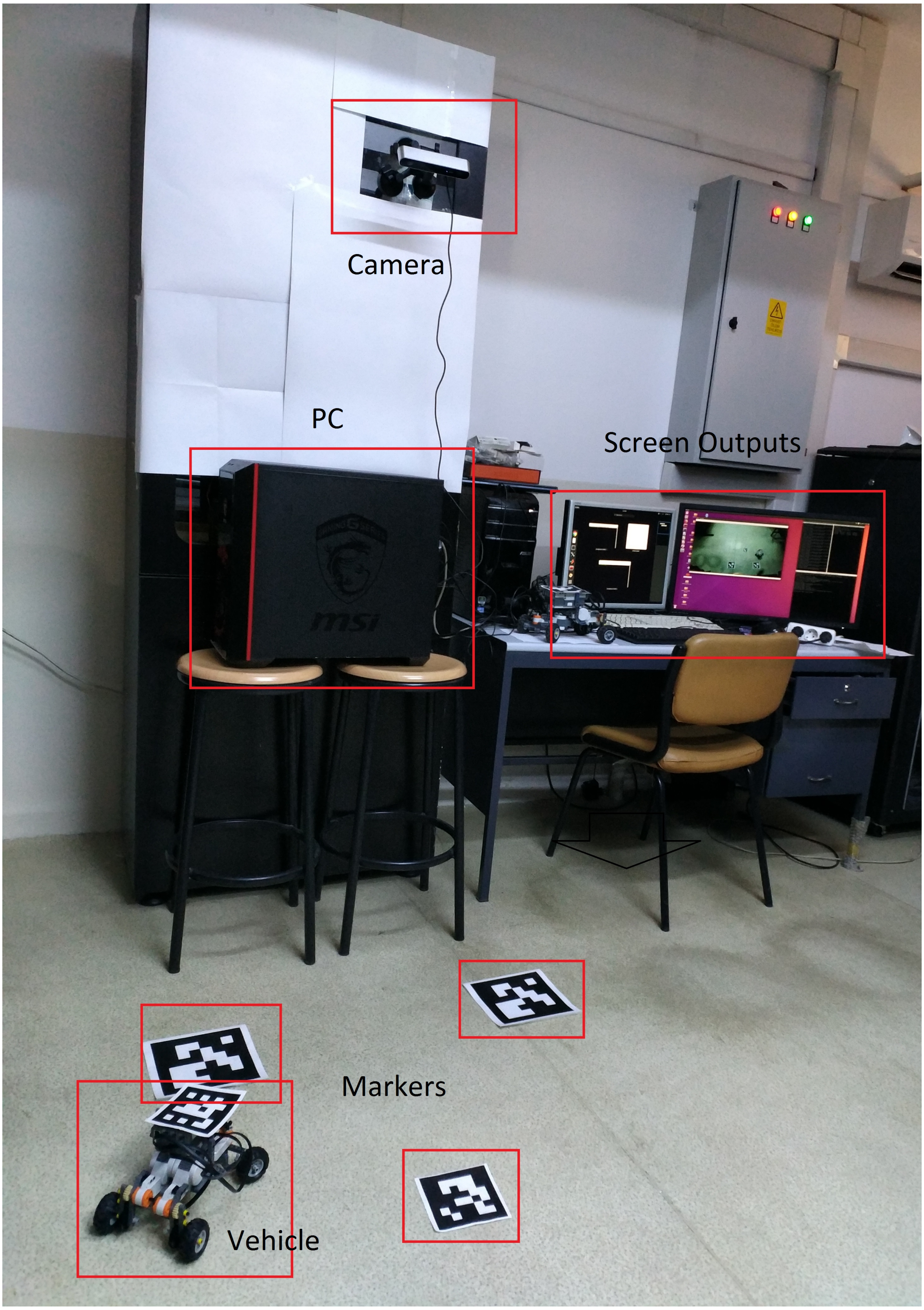

Figure 4.4: Experiment Environment

Figure 4.5: Camera and Screen Output

4.2 Experimental Scenarios

In this section, you can see screen outputs of each experimentation scenario as well as a close-up look on the generated direction and the path followed. There are two images on each screen output. The right one is the camera image of our experimentation environment. On the camera image, there is vehicle with a square marker on top of it and numerous square markers by themselves, representing obstacles. The left image is the output of our algorithm, which shows the obstacles, the vehicle and its orientation, the generated path (shown as blue line), and the followed path (shown as red line) as the vehicle moves forward. The last figure of each scenario shows the motion responses that are drawn using the logged data.

There are two obstacles on the bottom of the images in each scenario. Independent of the starting position and direction of the vehicle, the goal position and direction of the vehicle is between the obstacles and facing downward. In the Figure (a) of each scenario, the vehicle is at its initial position. In Figure (b), the vehicle is approximately halfway through the planned path. In Figure (c), the vehicle has completed its journey and arrived the goal. Figure (d) of each scenario shows a closer look at the experiment, drawn from using the logged data of the scenario.

4.2.1 Scenario 01

The experiments start with a simple case. The vehicle is straight above the goal position and does not required to change its direction.

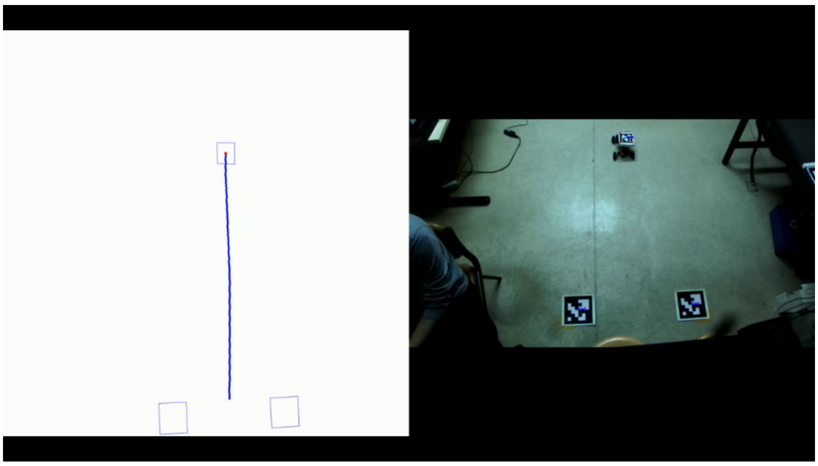

Figure 4.6: Scenario01-a

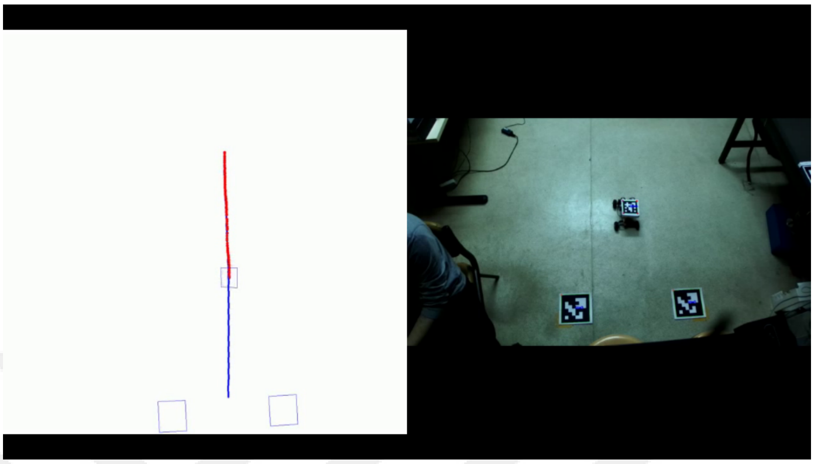

Figure 4.7: Scenario01-b

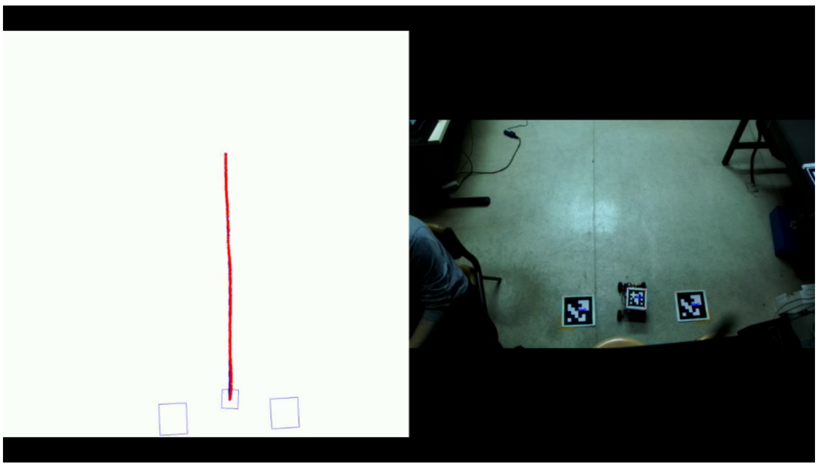

Figure 4.8: Scenario01-c

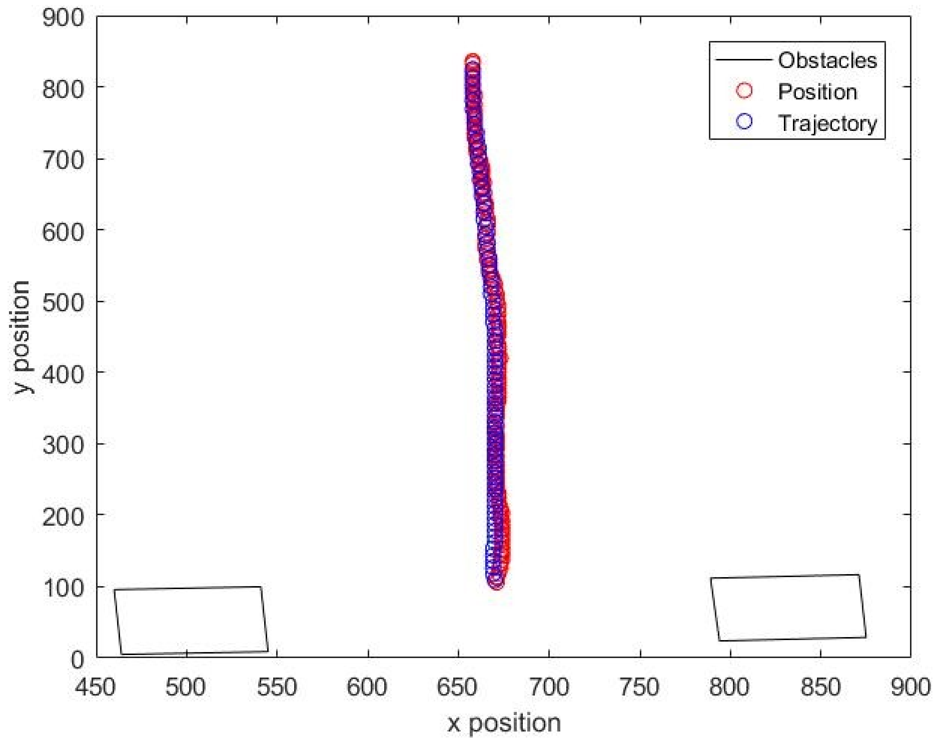

Figure 4.9: Scenario01-d

4.2.2 Scenario 02

In this scenario, the vehicle is on the right side of the experimentation area.

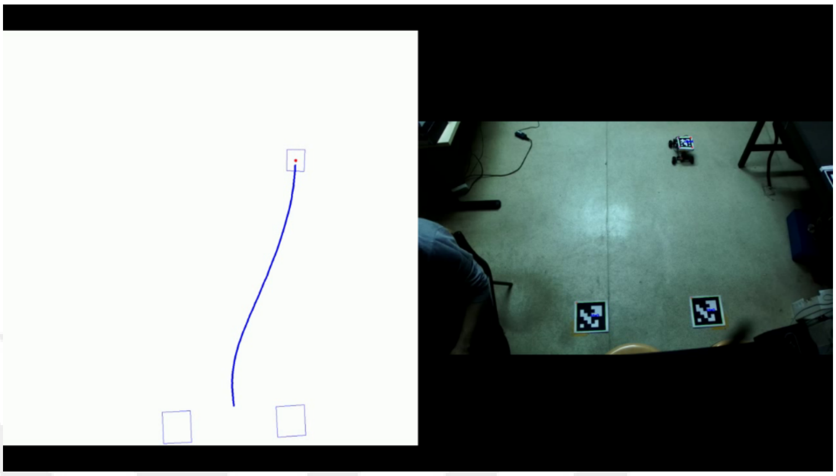

Figure 4.10: Scenario02-a

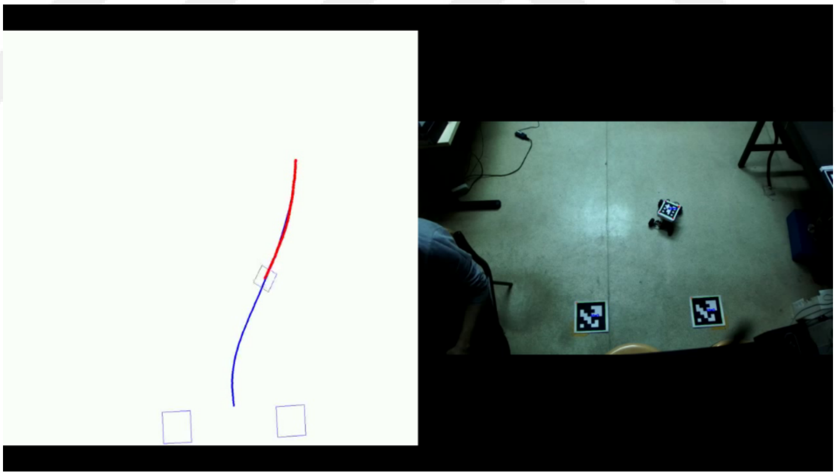

Figure 4.11: Scenario02-b

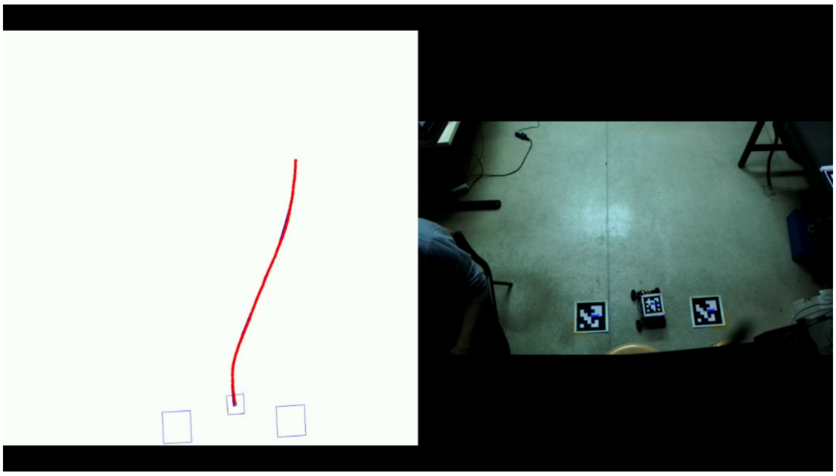

Figure 4.12: Scenario02-c

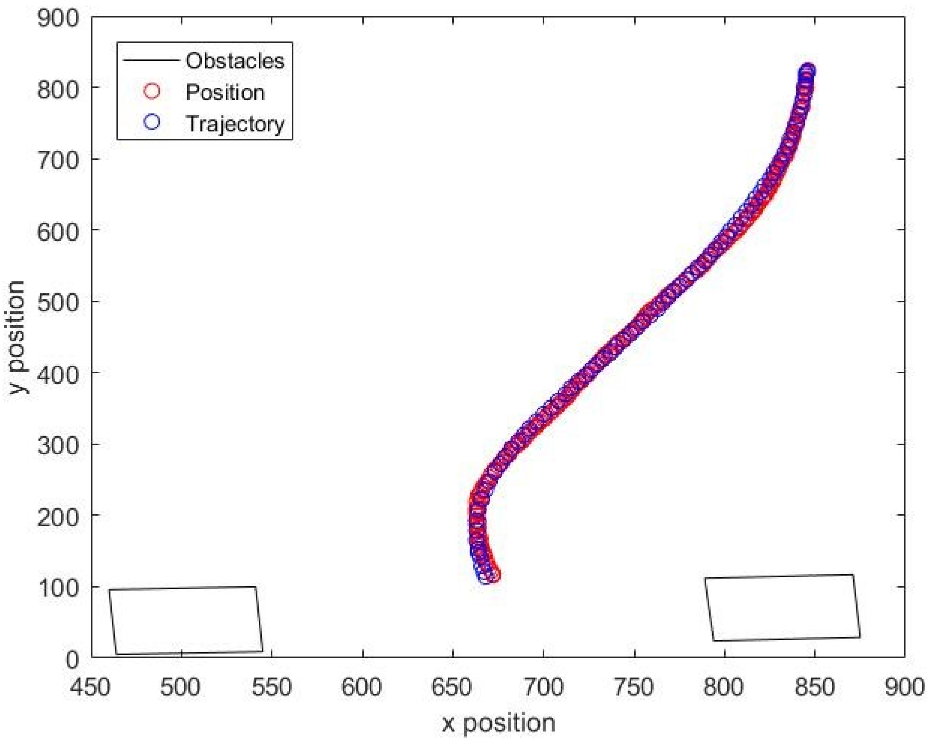

Figure 4.13: Scenario02-d

4.2.3 Scenario 03

This scenario is mirror image of Scenario02.

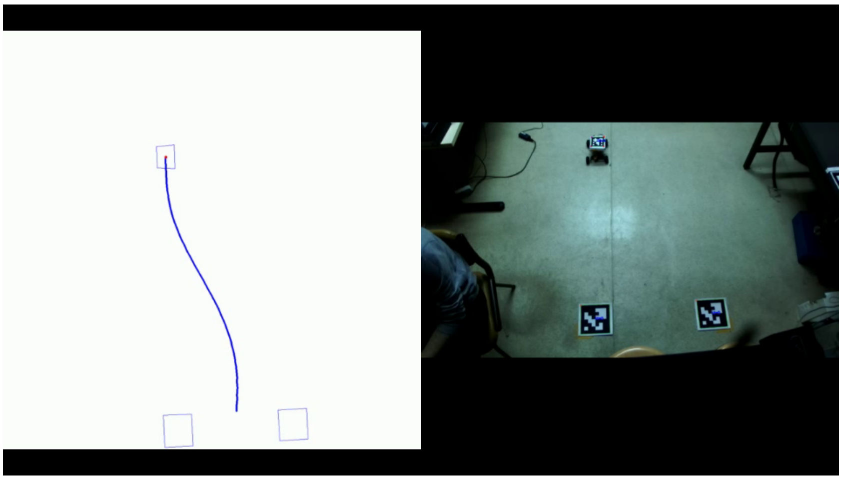

Figure 4.14: Scenario03-a

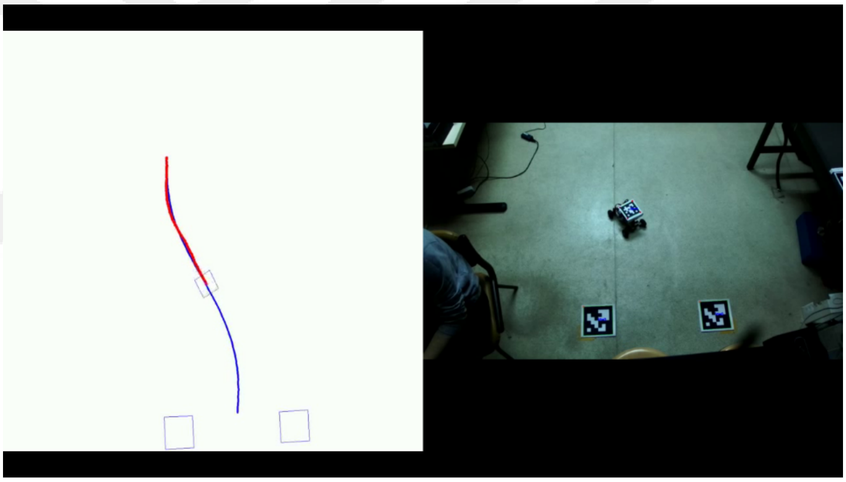

Figure 4.15: Scenario03-b

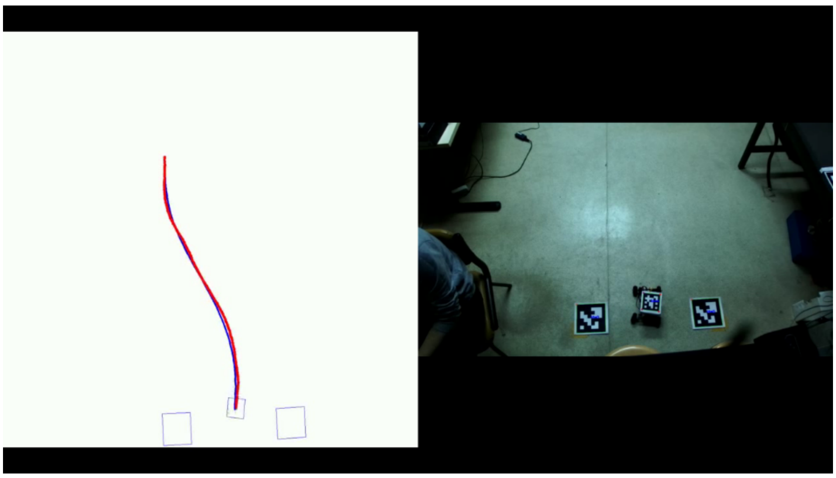

Figure 4.16: Scenario03-c

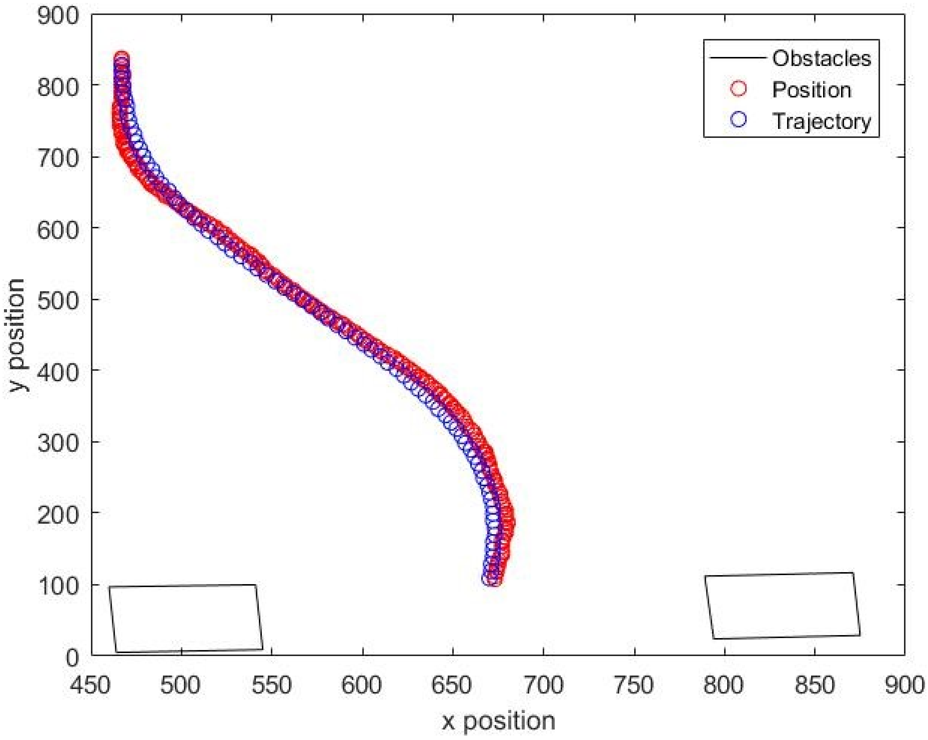

Figure 4.17: Scenario03-d

4.2.4 Scenario 04

In this scenario the starting position and direction of the vehicle is the same with Scenario01 however, there is an additional obstacle in front of the vehicle. The vehicle re-routes towards the end of its journey.

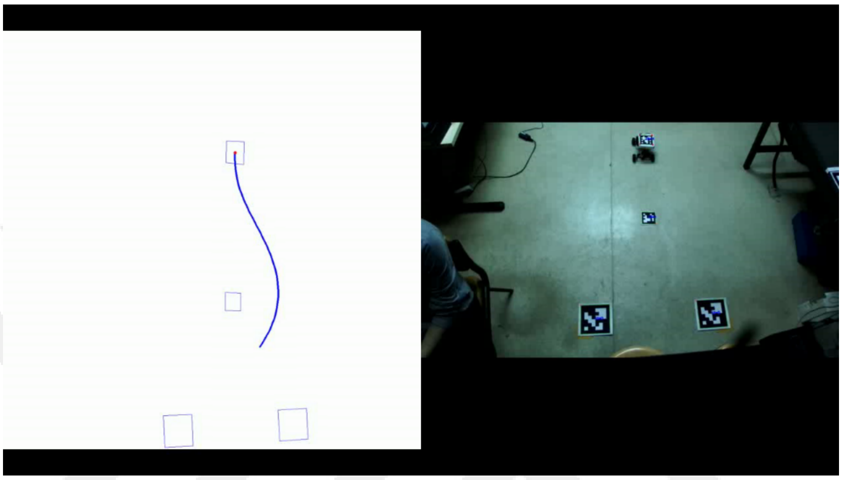

Figure 4.18: Scenario04-a

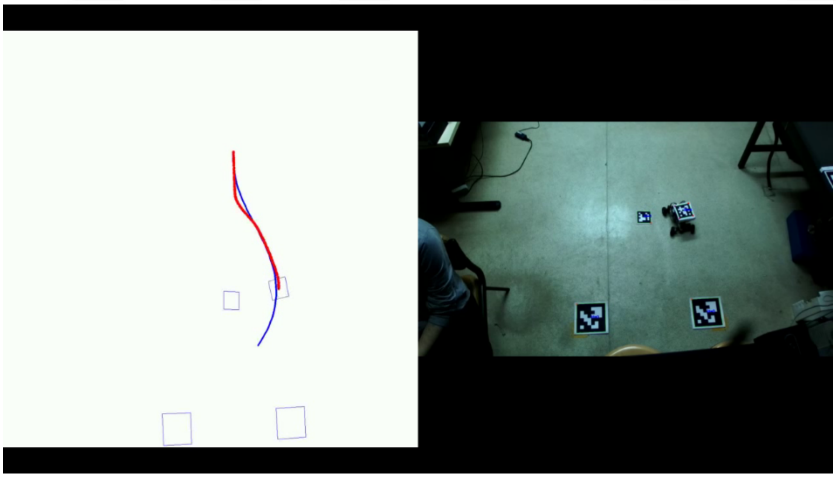

Figure 4.19: Scenario04-b

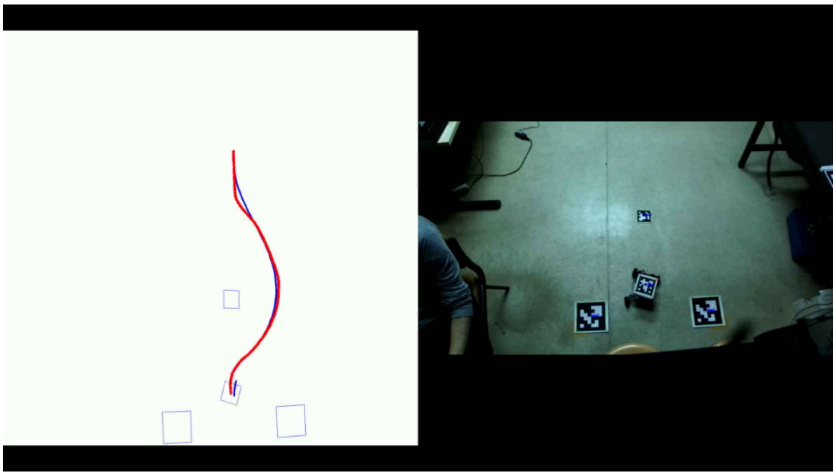

Figure 4.20: Scenario04-c

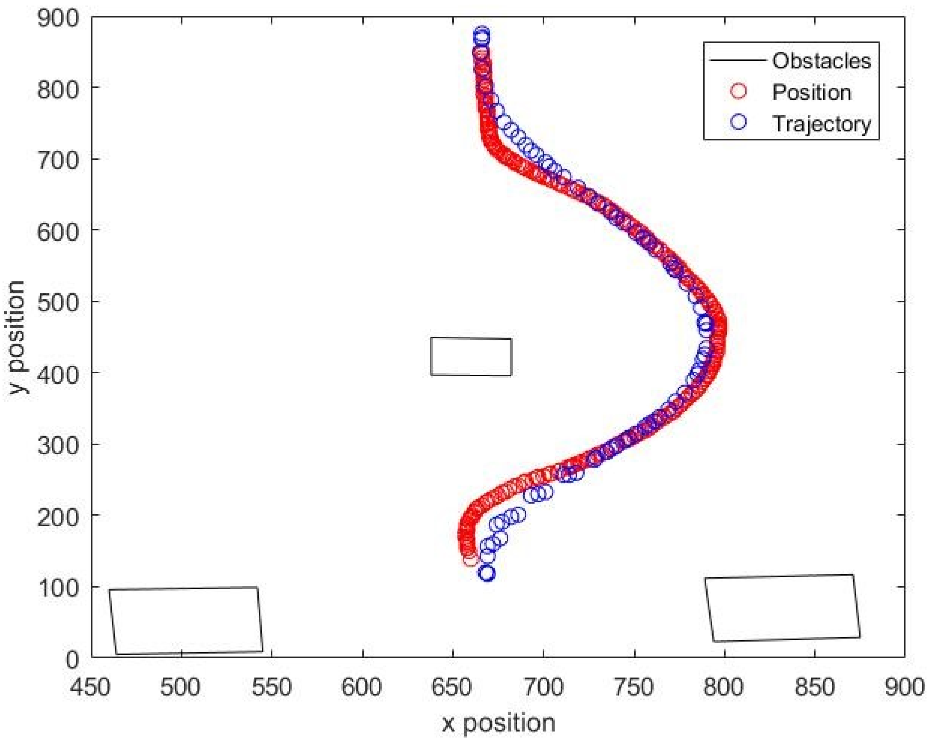

Figure 4.21: Scenario04-d

4.2.5 Scenario 05

This scenario is similar to Scenario04. Only difference is the starting position of the vehicle, which is to right.

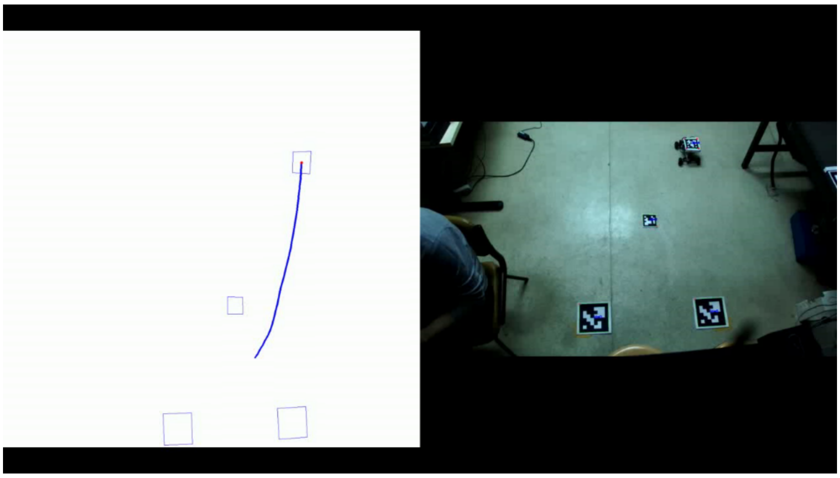

Figure 4.22: Scenario05-a

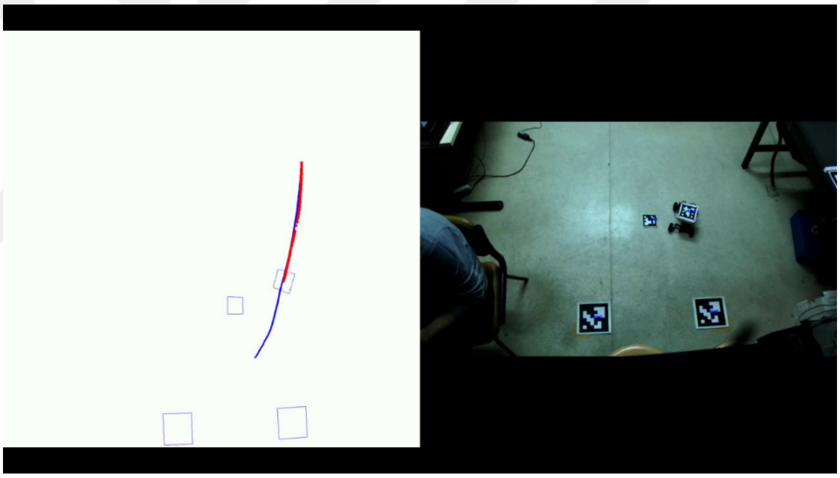

Figure 4.23: Scenario05-b

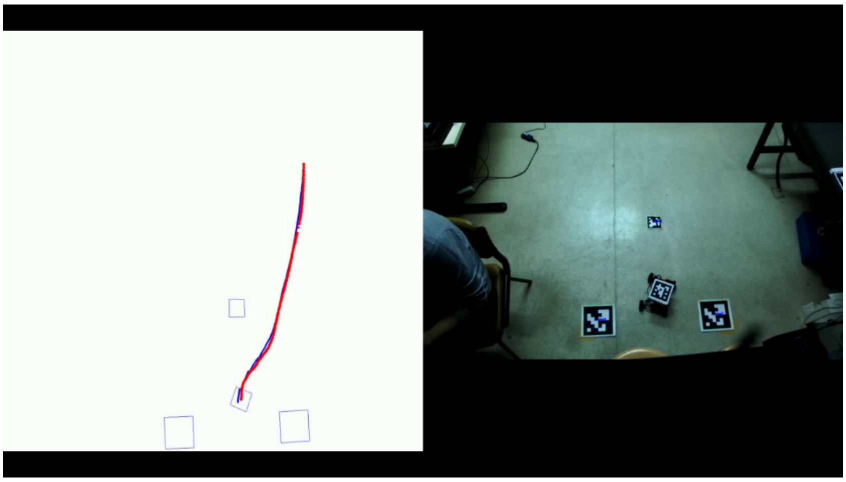

Figure 4.24: Scenario05-c

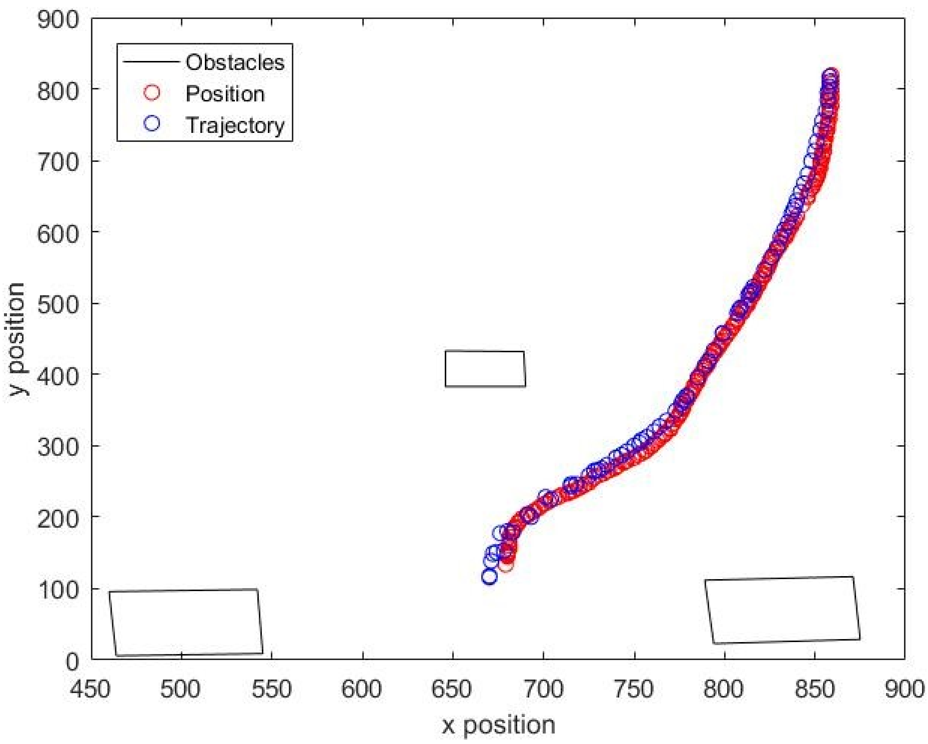

Figure 4.25: Scenario05-d

4.2.6 Scenario 06

This scenario is mirror image of Scenario05. The vehicle re-routes a few times.

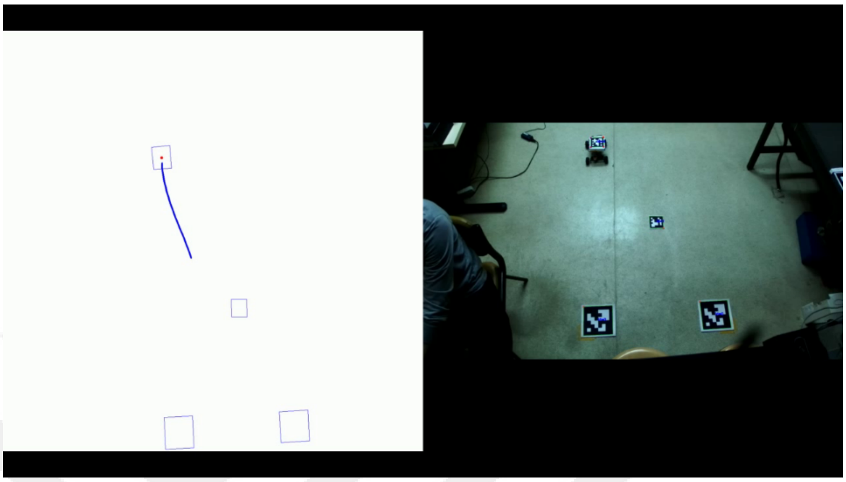

Figure 4.26: Scenario06-a

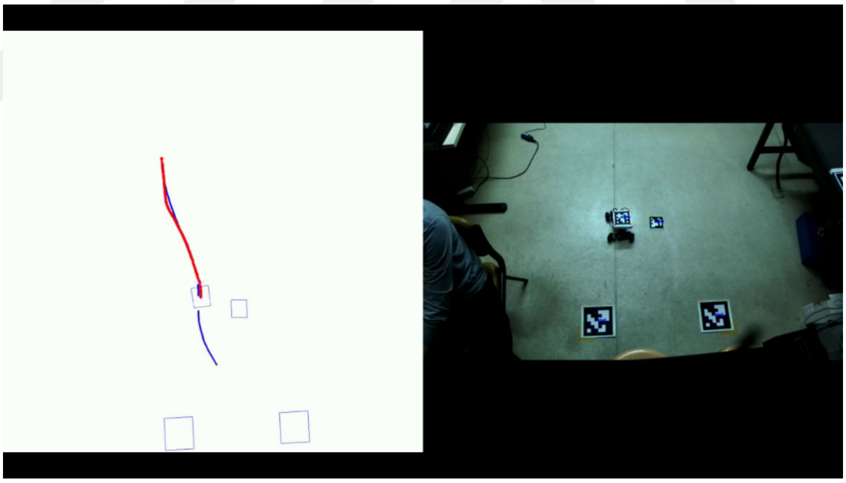

Figure 4.27: Scenario06-b

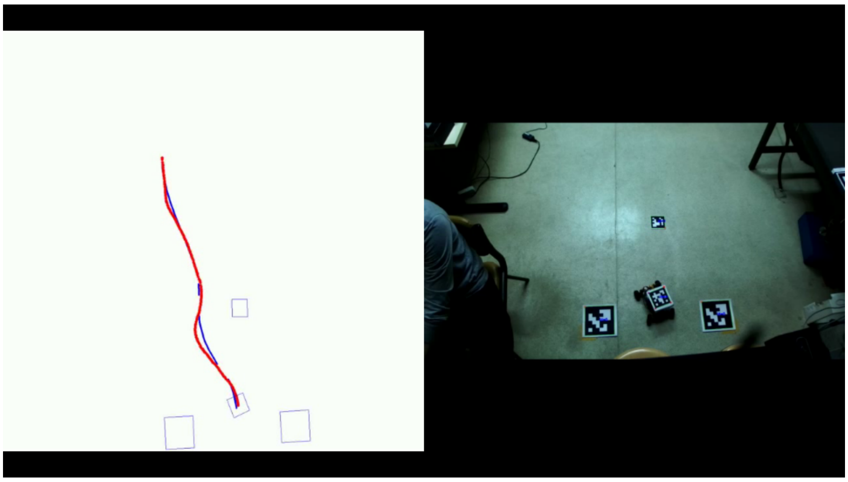

Figure 4.28: Scenario06-c

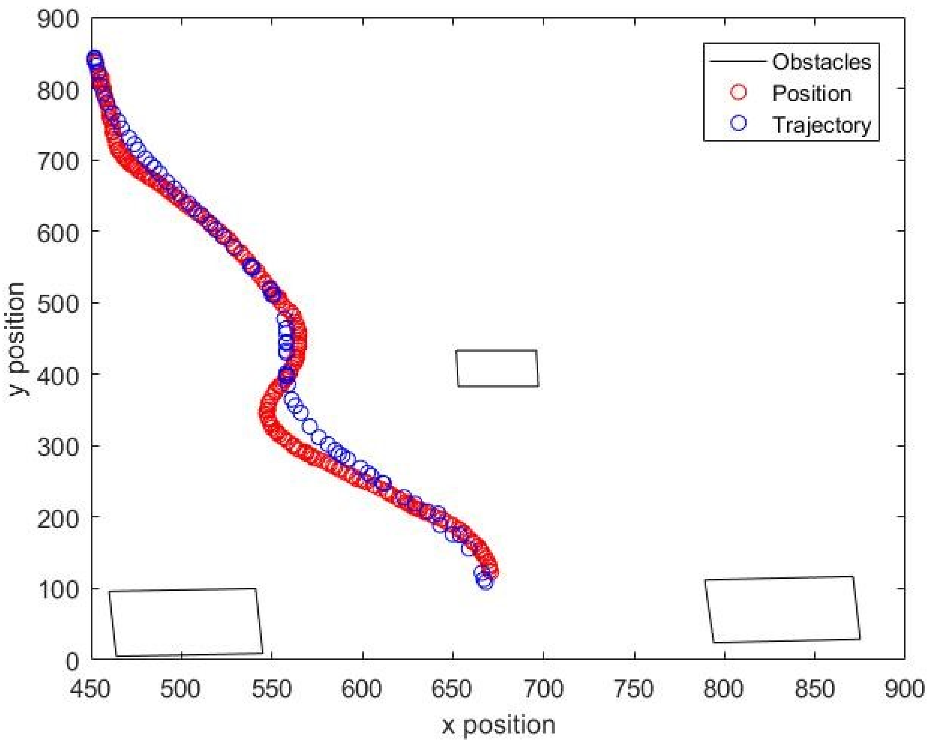

Figure 4.29: Scenario06-d

4.2.7 Scenario 07

This scenario is a version of Scenario01 in which the vehicle's direction is to the right. The vehicle re-routes a few times in this scenario.

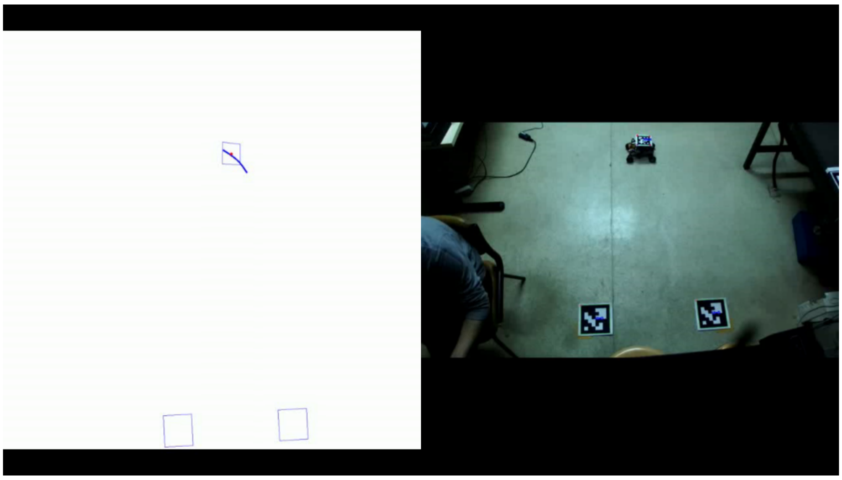

Figure 4.30: Scenario07-a

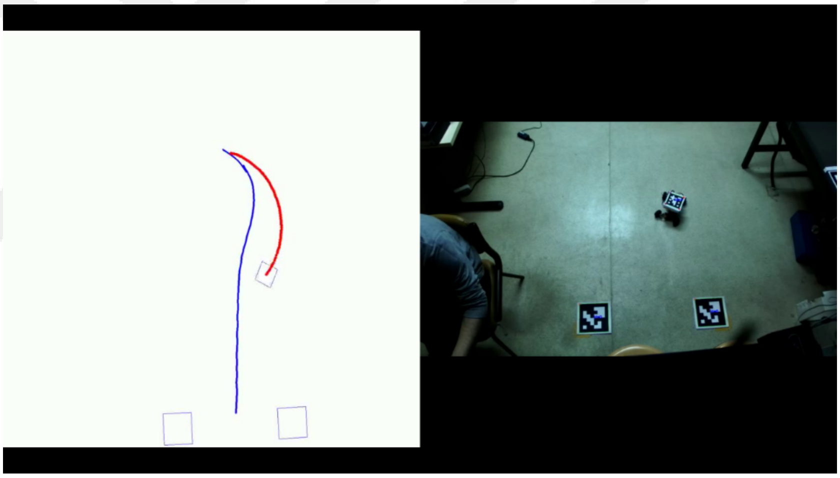

Figure 4.31: Scenario07-b

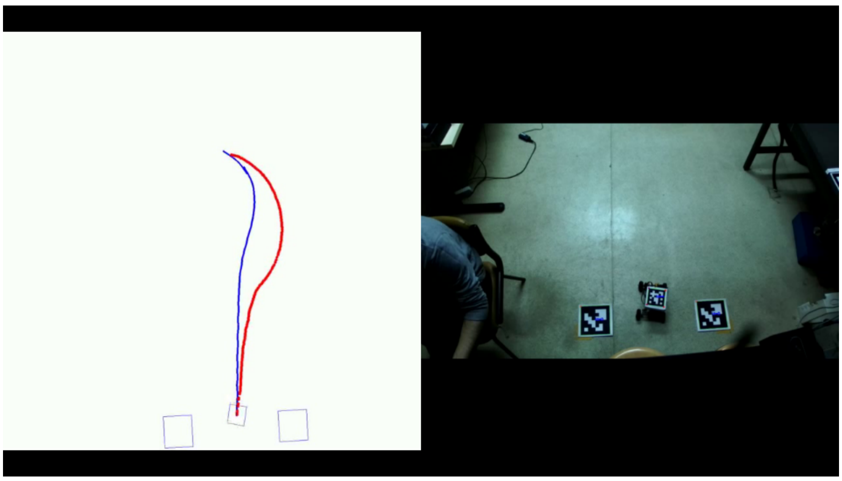

Figure 4.32: Scenario07-c

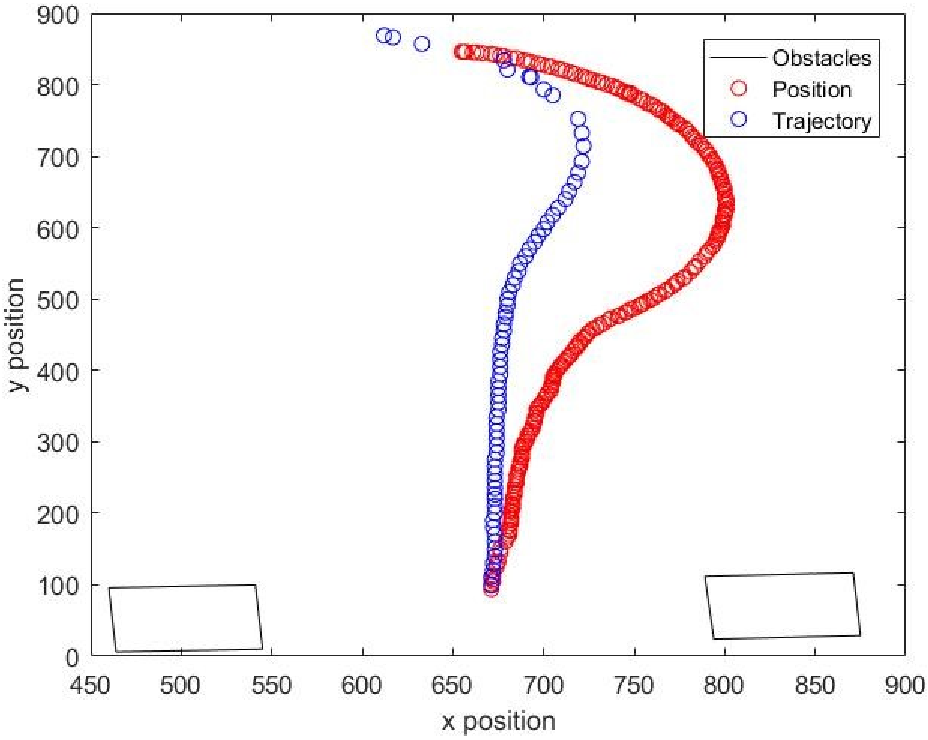

Figure 4.33: Scenario07-d

4.2.8 Scenario 08

This scenario is mirror image of Scenario07, meaning the vehicle is facing left. The vehicle re-routes towards the end of its journey.

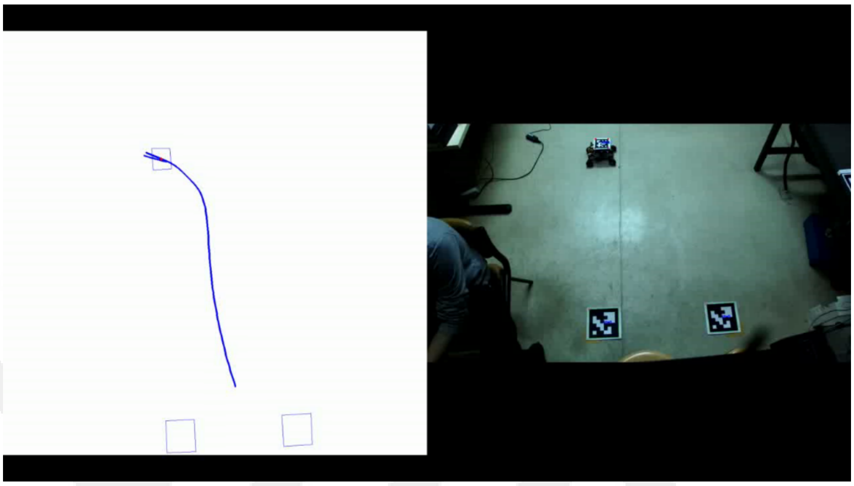

Figure 4.34: Scenario08-a

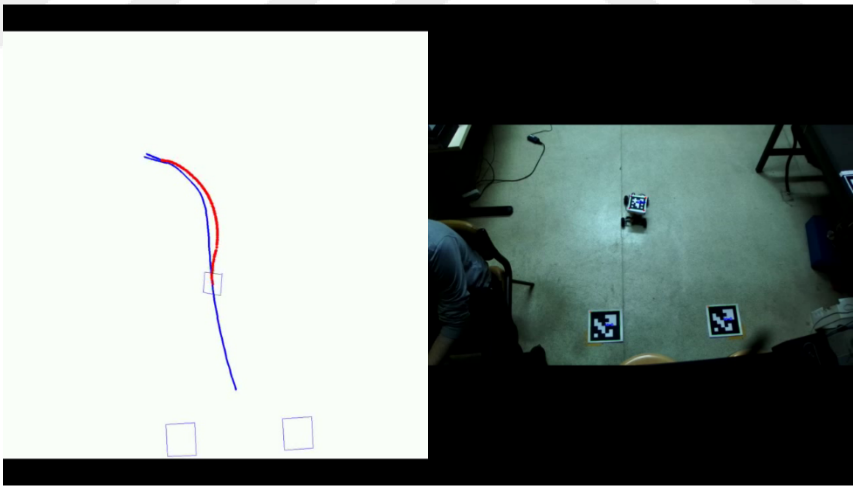

Figure 4.35: Scenario08-b

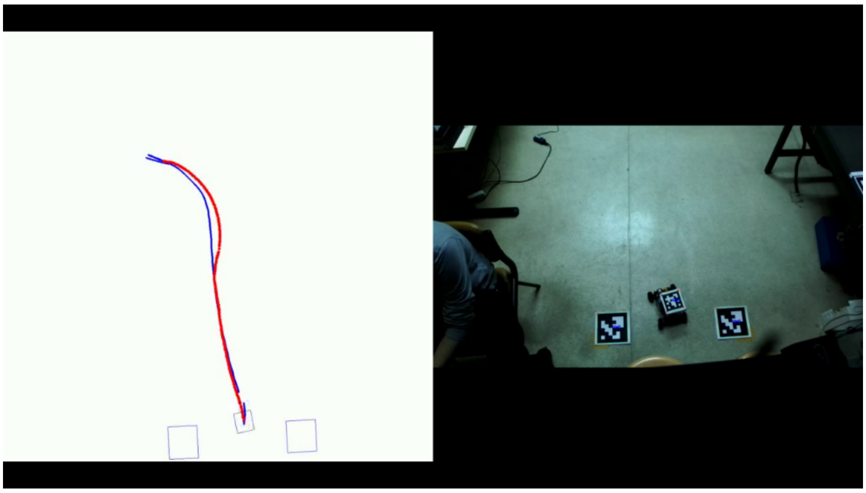

Figure 4.36: Scenario08-c

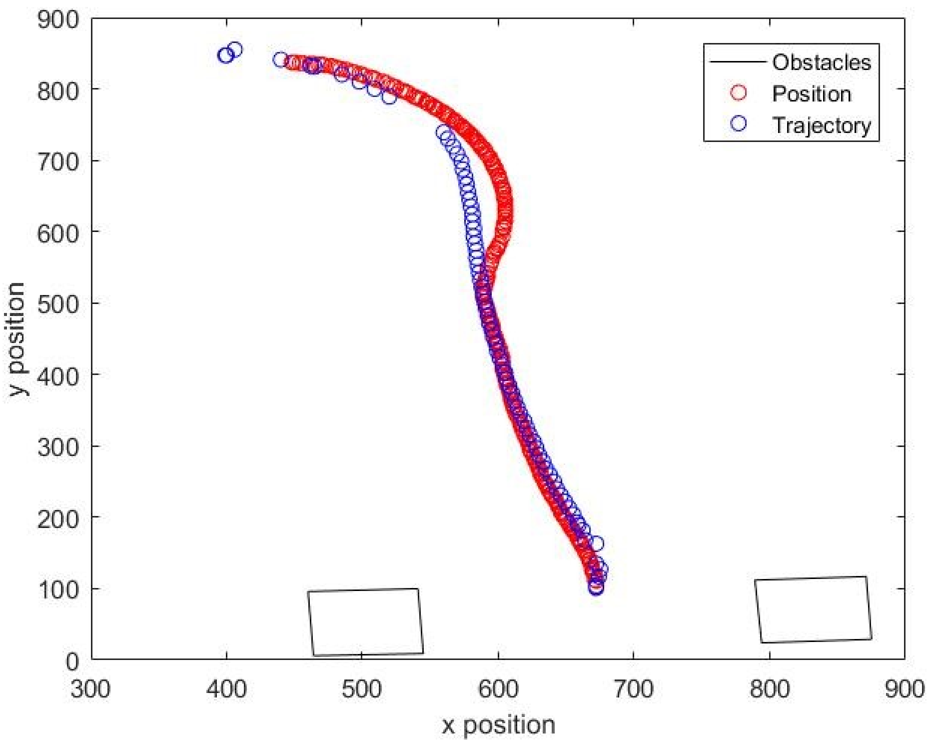

Figure 4.37: Scenario08-d

4.2.9 Scenario 09

This scenario is a version of Scenario07 with the vehicle's starting direction is to the left.

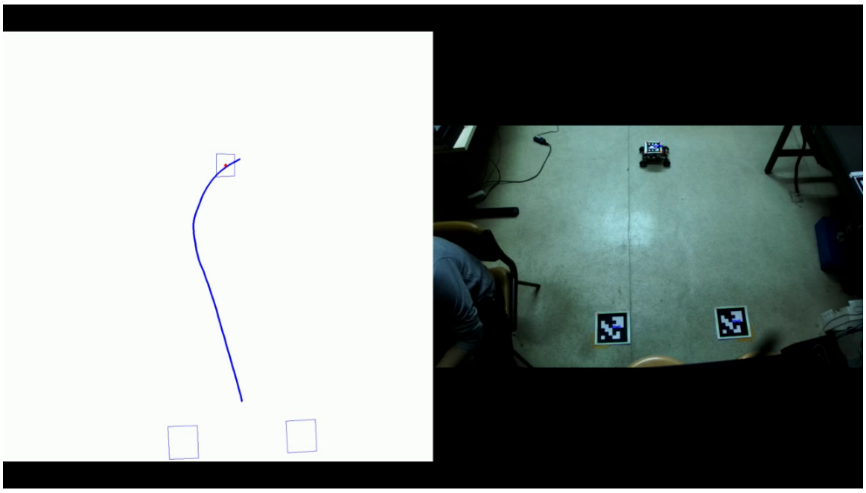

Figure 4.38: Scenario09-a

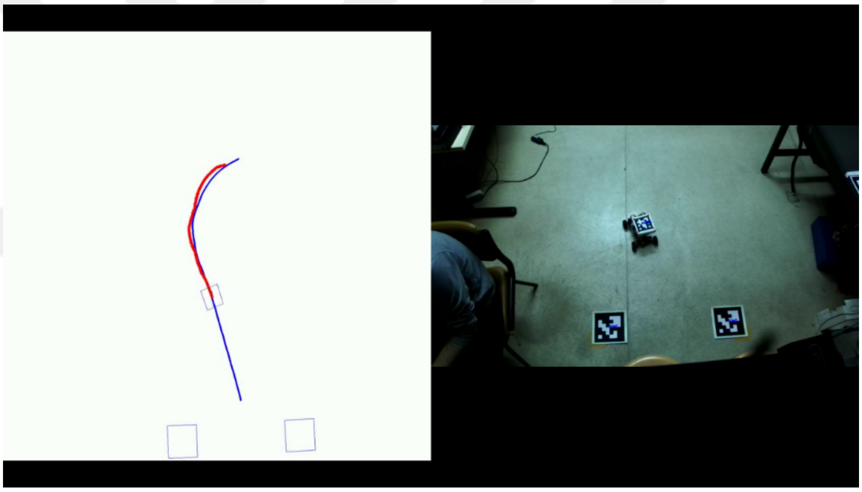

Figure 4.39: Scenario09-b

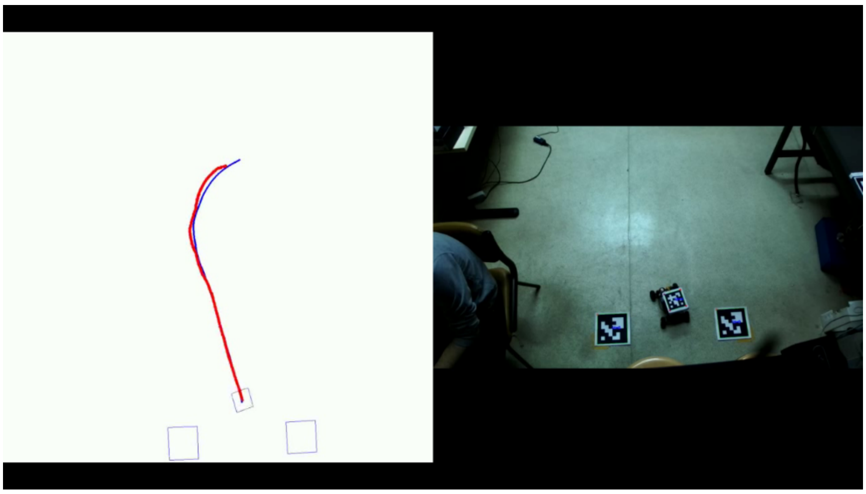

Figure 4.40: Scenario09-c

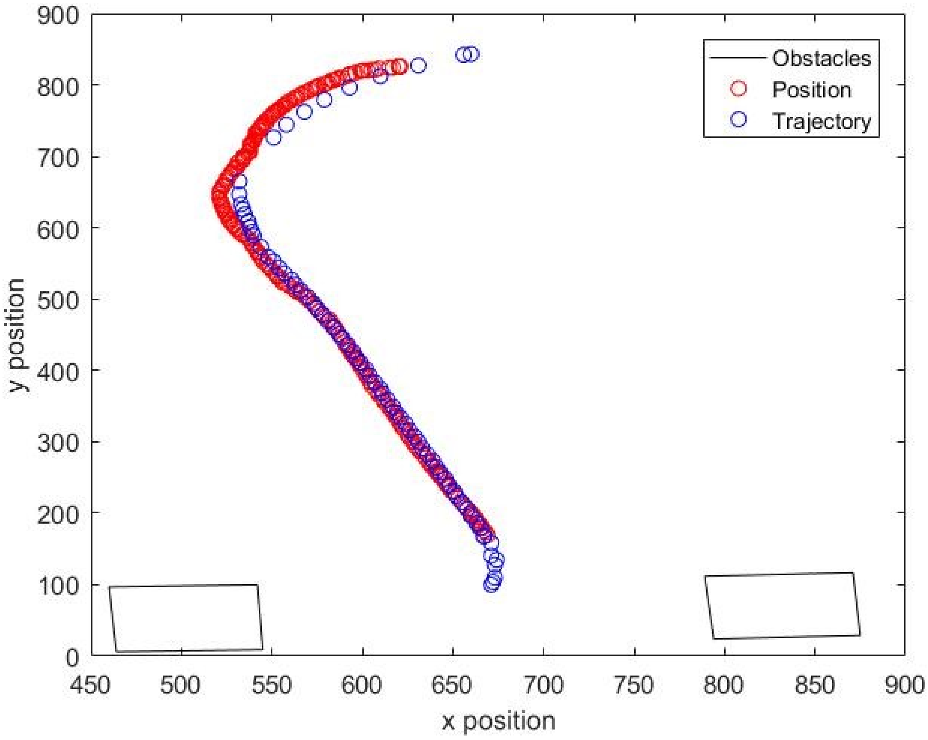

Figure 4.41: Scenario09-d

4.2.10 Scenario 10

This scenario is a version of Scenario09 with the vehicle's starting position is on the right side.

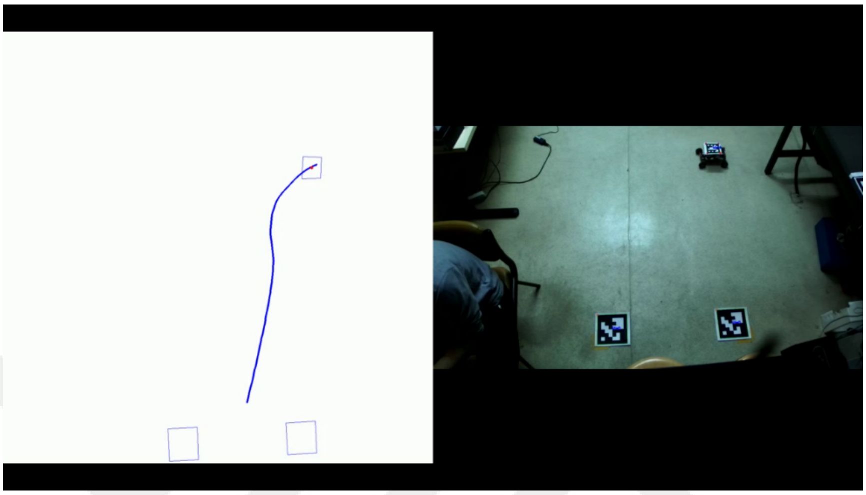

Figure 4.42: Scenario10-a

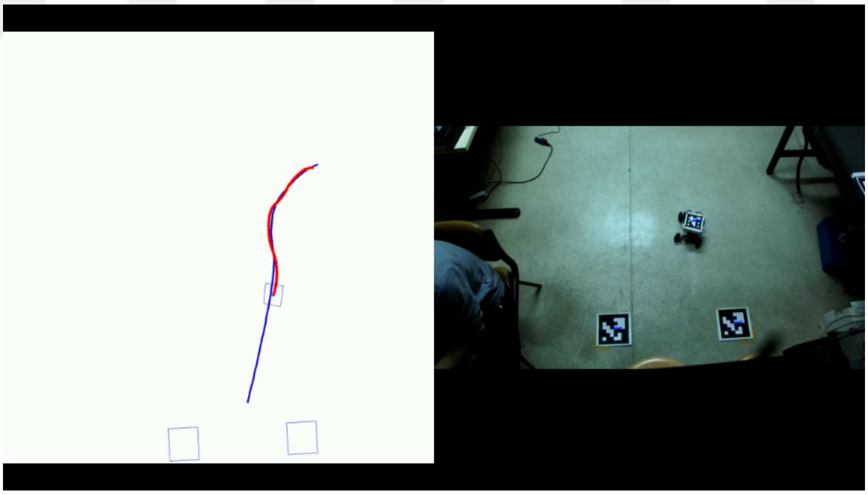

Figure 4.43: Scenario10-b

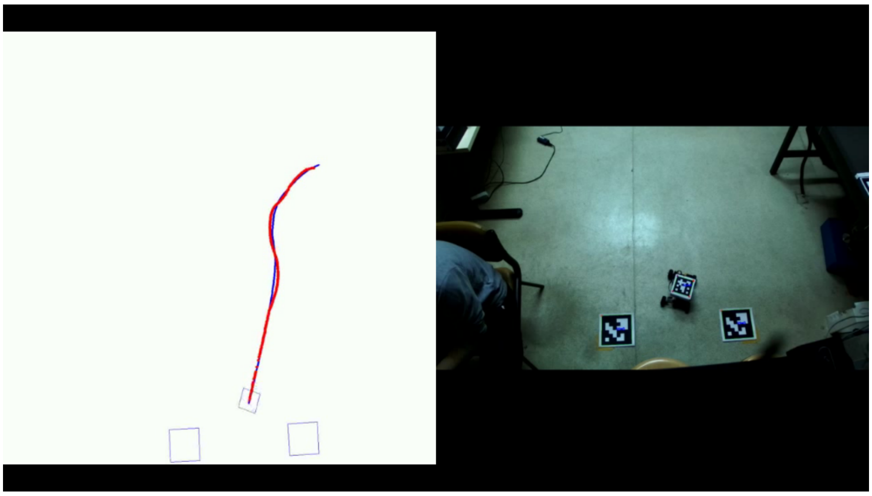

Figure 4.44: Scenario10-c

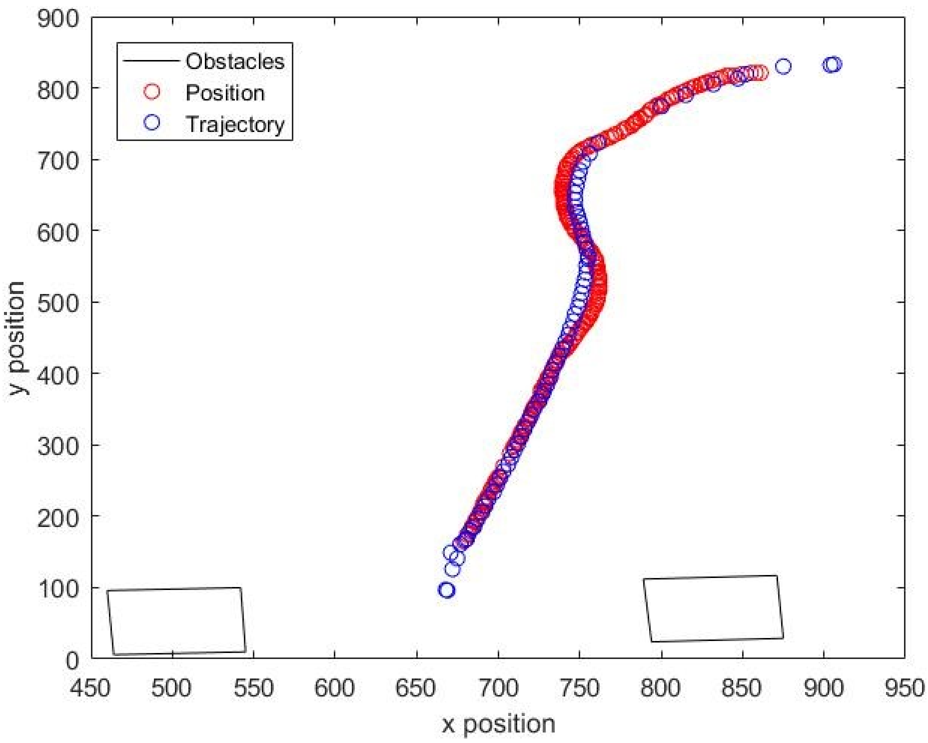

Figure 4.45: Scenario10-d

4.2.11 Scenario 11

In this scenario, the vehicle's starting position is much closer to the destination compared to the previous scenarios, it is on the left side of the experimentation area, facing right side. Also there is an additional obstacle. The vehicle re-routes a few times in this scenario.

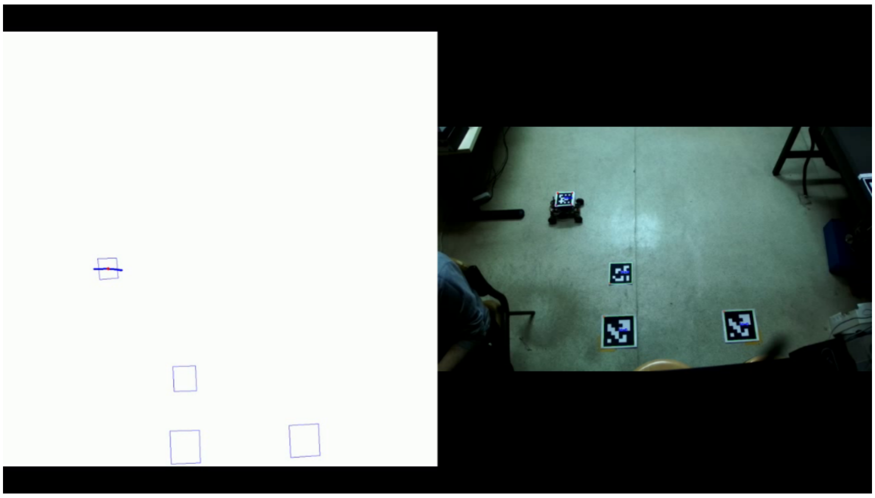

Figure 4.46: Scenario11-a

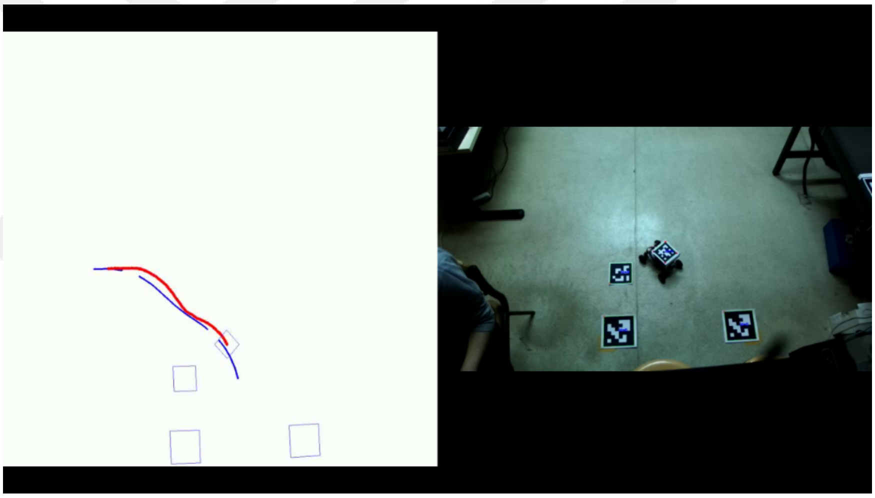

Figure 4.47: Scenario11-b

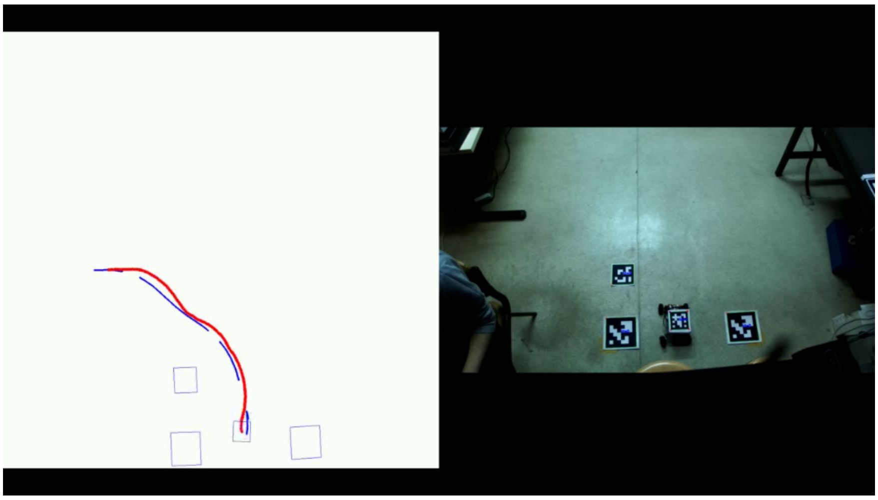

Figure 4.48: Scenario11-c

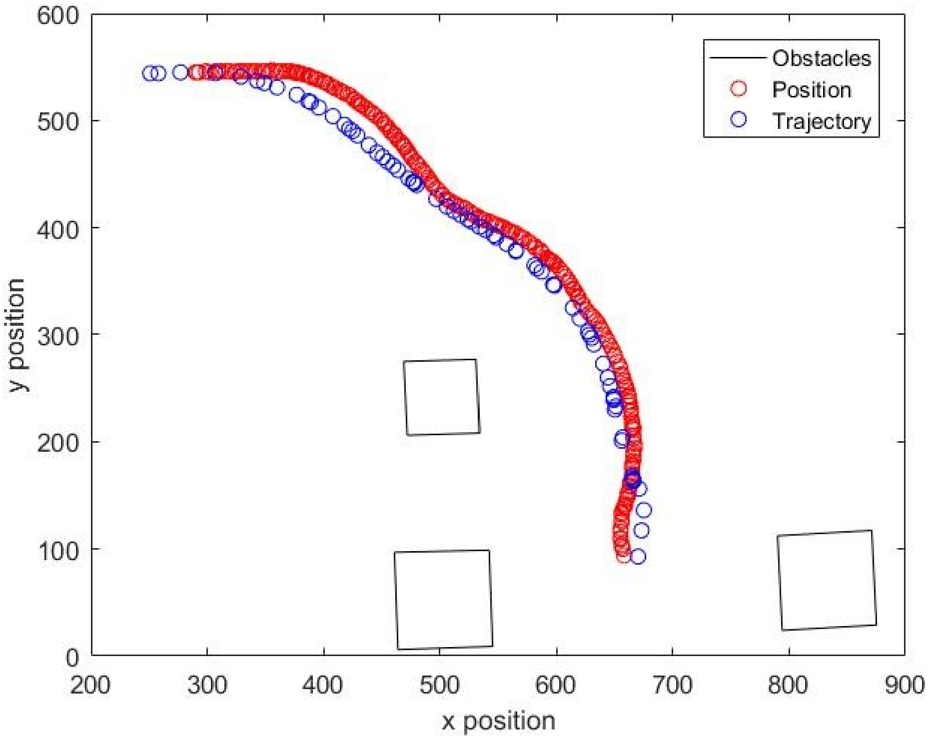

Figure 4.49: Scenario11-d

4.2.12 Scenario 12

This scenario is mirror image of Scenario11.

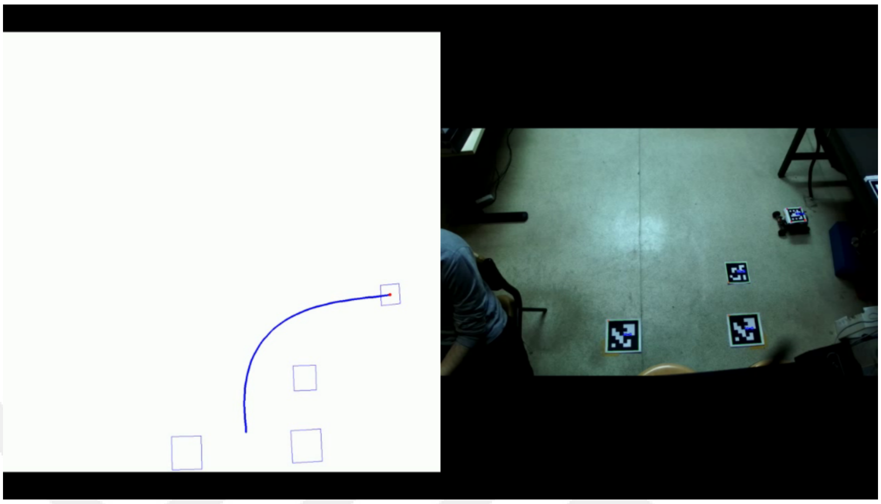

Figure 4.50: Scenario12-a

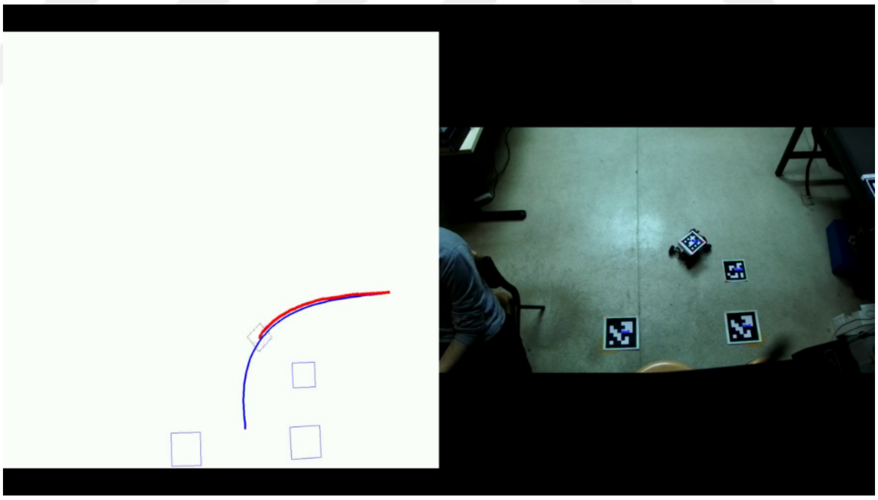

Figure 4.51: Scenario12-b

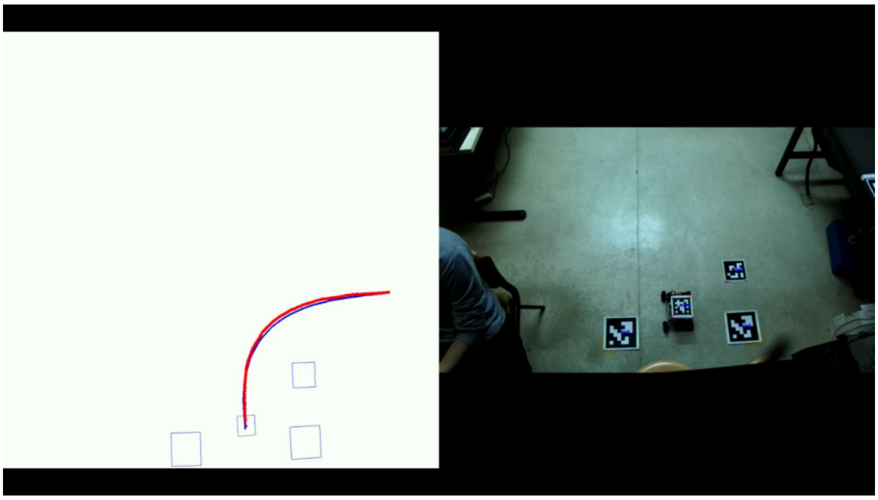

Figure 4.52: Scenario12-c

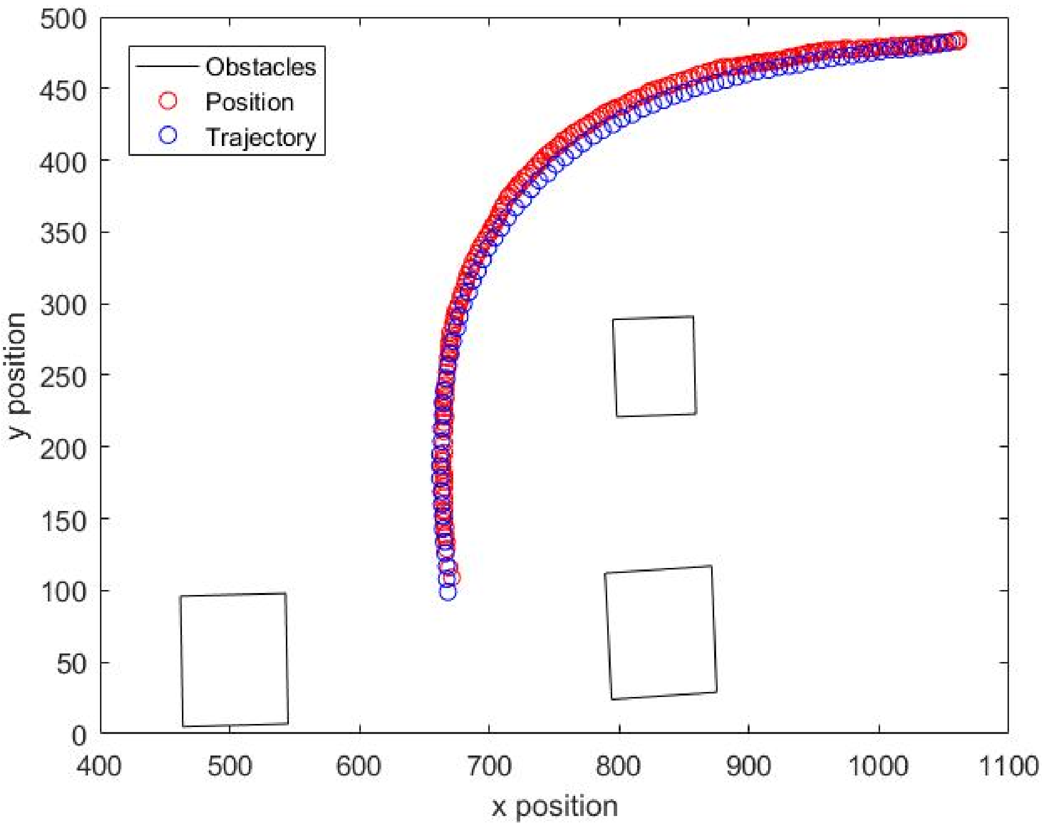

Figure 4.53: Scenario12-d

4.2.13 Scenario 13

In this scenario, the vehicle has identical starting position with Scenario 11, but now the vehicle faces downwards and there are no additional obstacles. The vehicle re-routes a few times in this scenario.

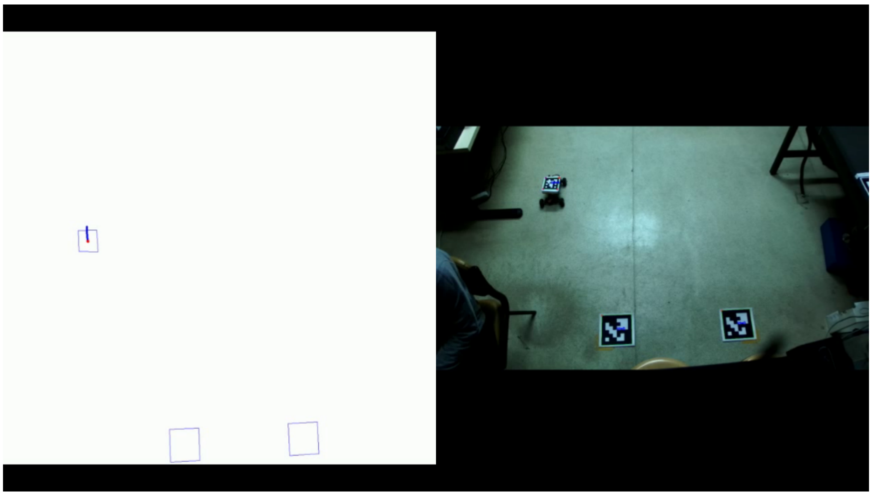

Figure 4.54: Scenario13-a

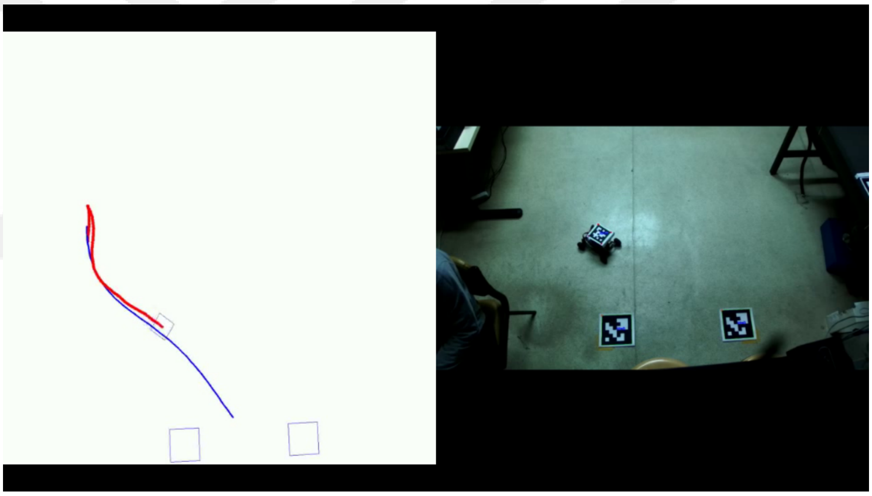

Figure 4.55: Scenario13-b

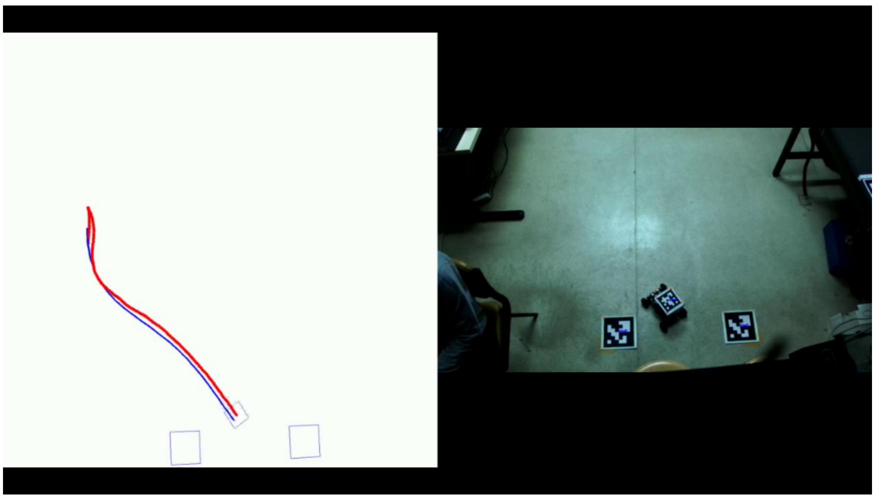

Figure 4.56: Scenario13-c

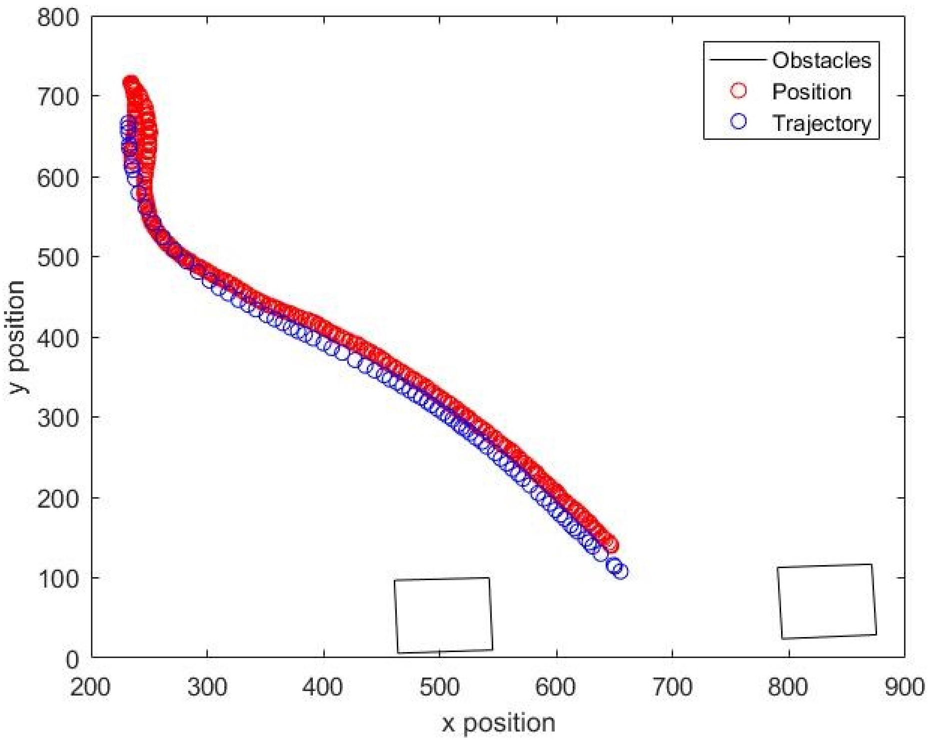

Figure 4.57: Scenario13-d

4.2.14 Scenario 14

This scenario is mirror image of Scenario13. The vehicle re-routes once in the beginning of its journey.

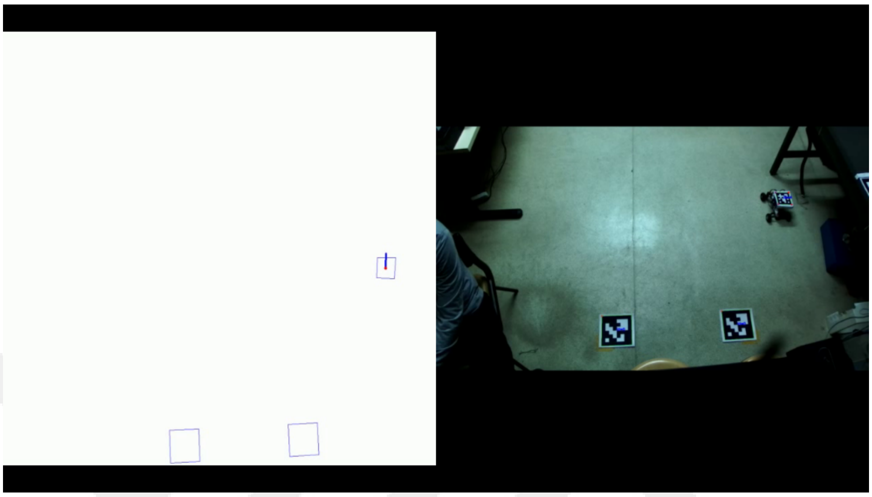

Figure 4.58: Scenario14-a

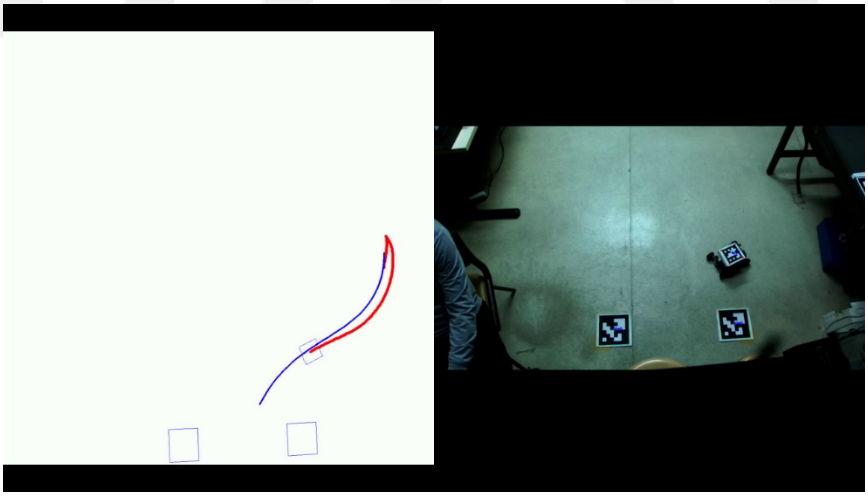

Figure 4.59: Scenario14-b

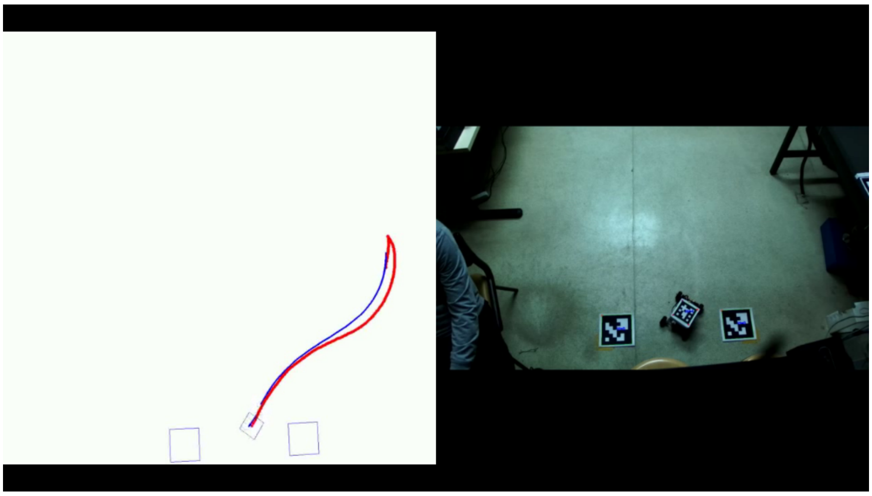

Figure 4.60: Scenario14-c

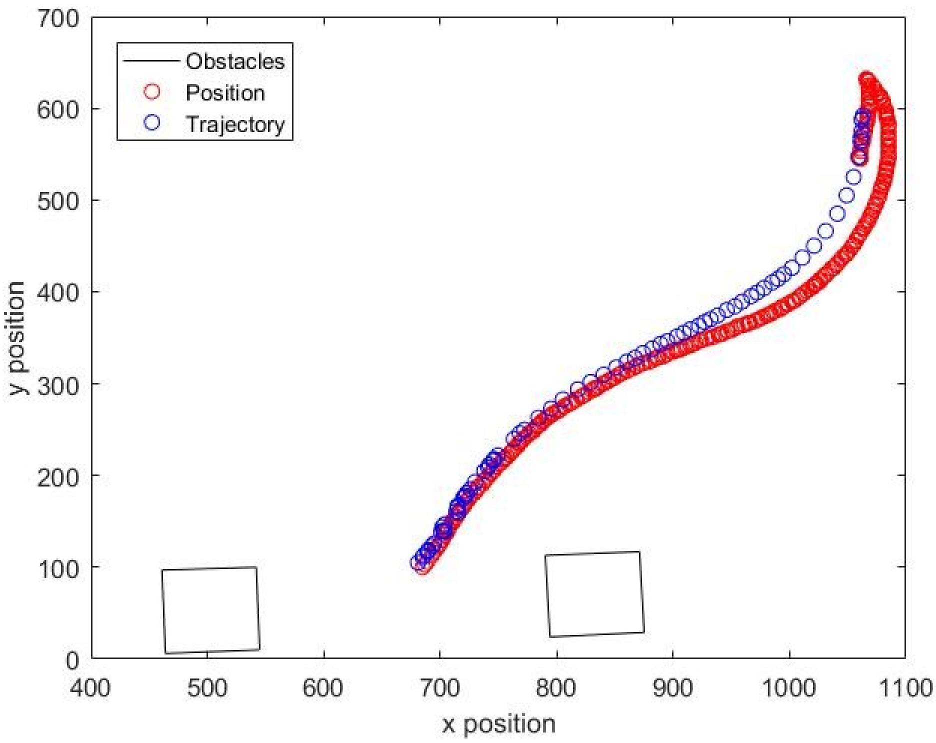

Figure 4.61: Scenario14-d

4.2.15 Scenario 15

In this scenario, the vehicle starts close to the destination and facing upwards. The vehicle re-routes a few times in this scenario.

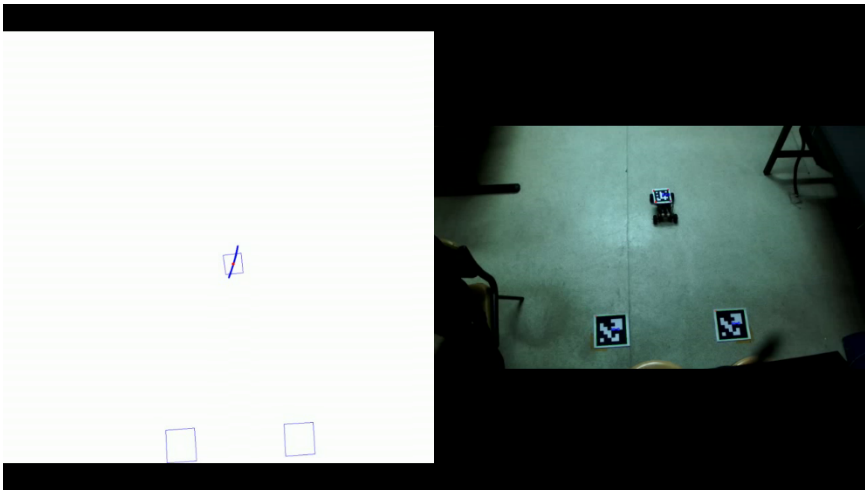

Figure 4.62: Scenario15-a

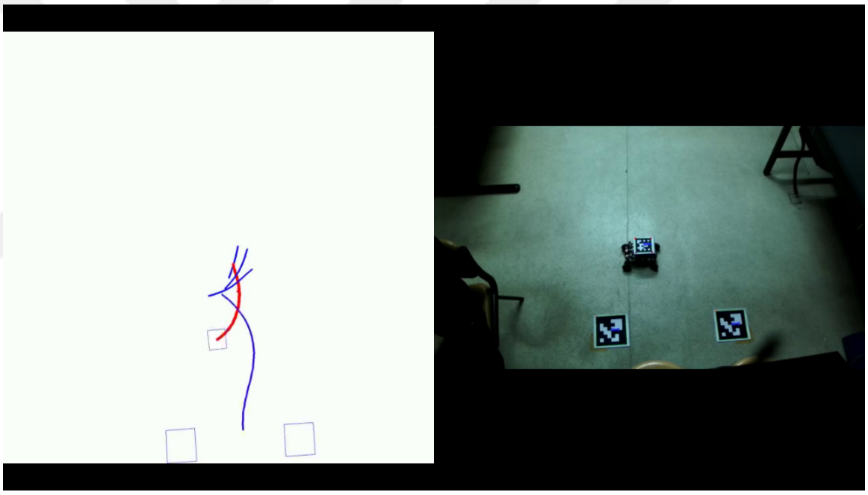

Figure 4.63: Scenario15-b

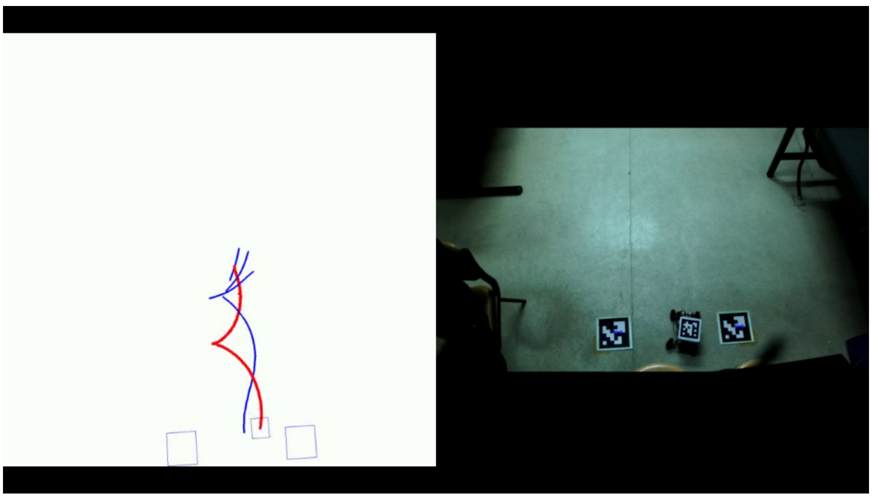

Figure 4.64: Scenario15-c

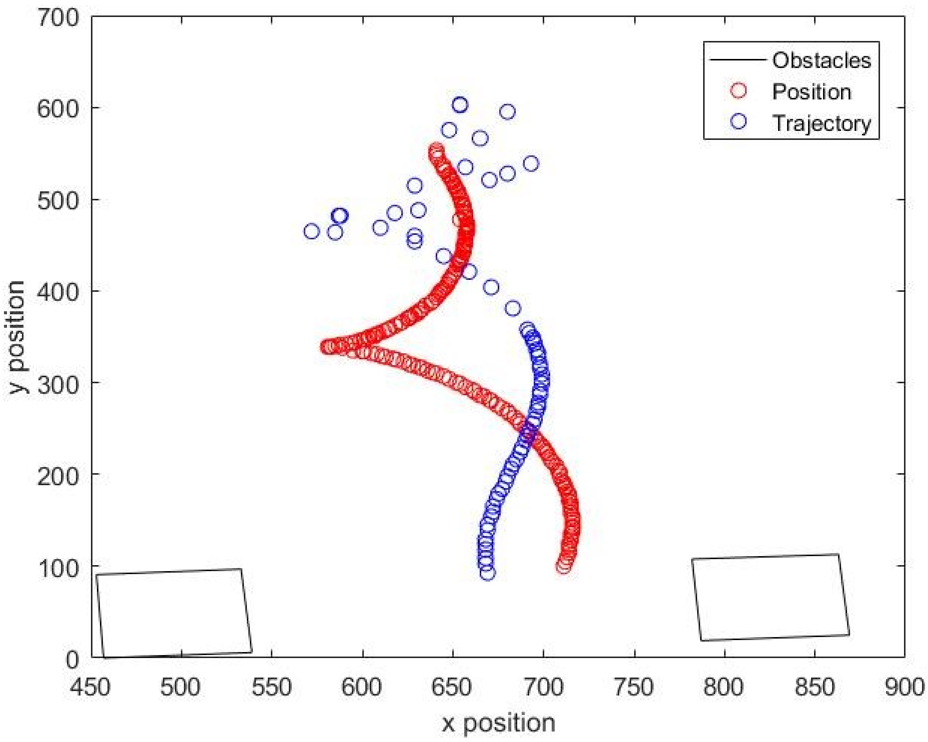

Figure 4.65: Scenario15-d

4.2.16 Scenario 16

This scenario is similar to Scenario15. The only difference is the vehicle starts from the right side on this scenario. The vehicle re-routes once in the beginning of its journey.

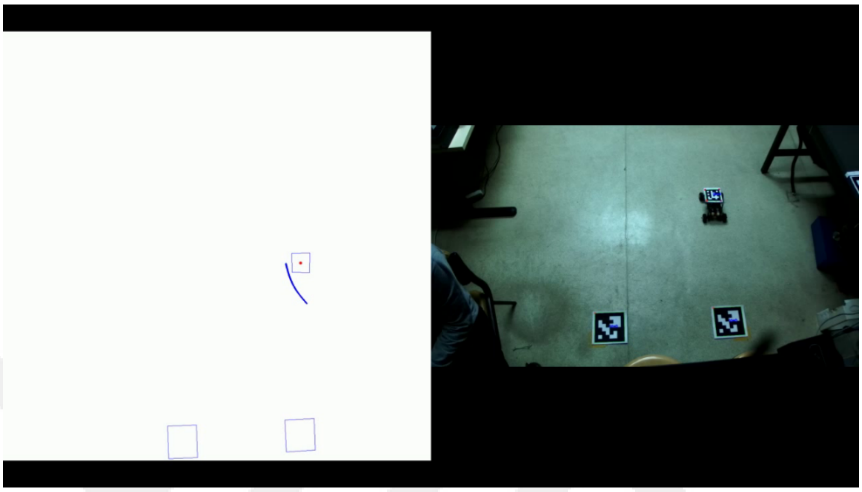

Figure 4.66: Scenario16-a

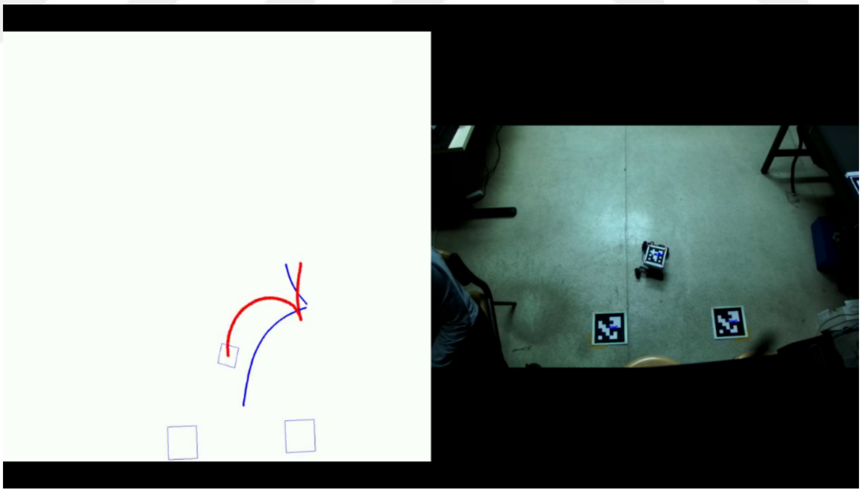

Figure 4.67: Scenario16-b

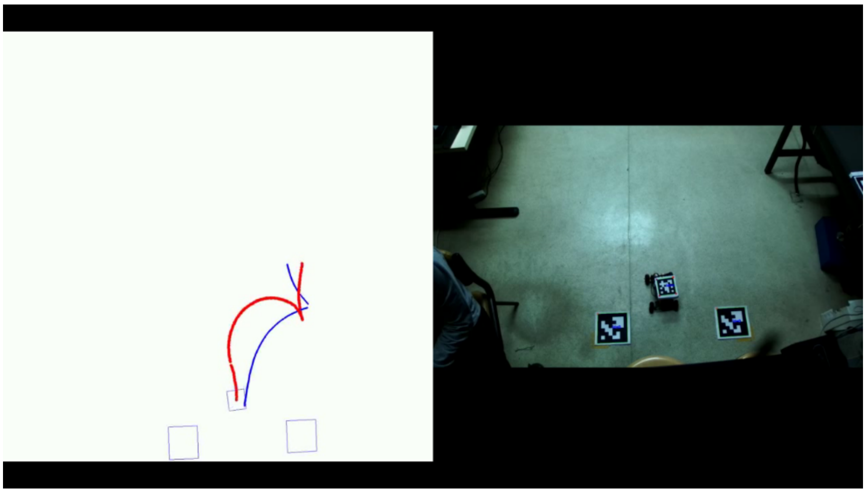

Figure 4.68: Scenario16-c

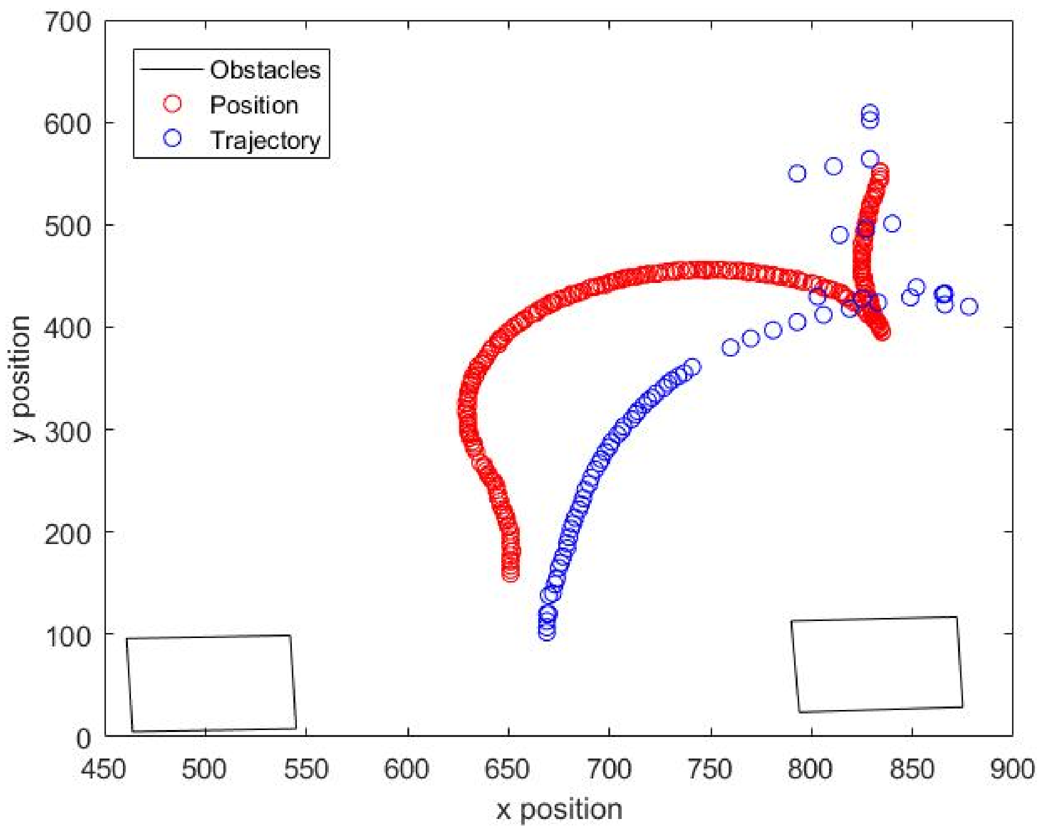

Figure 4.69: Scenario16-d

4.2.17 Scenario 17

This scenario is mirror image of Scenario16. The vehicle re-routes once in the beginning of its journey.

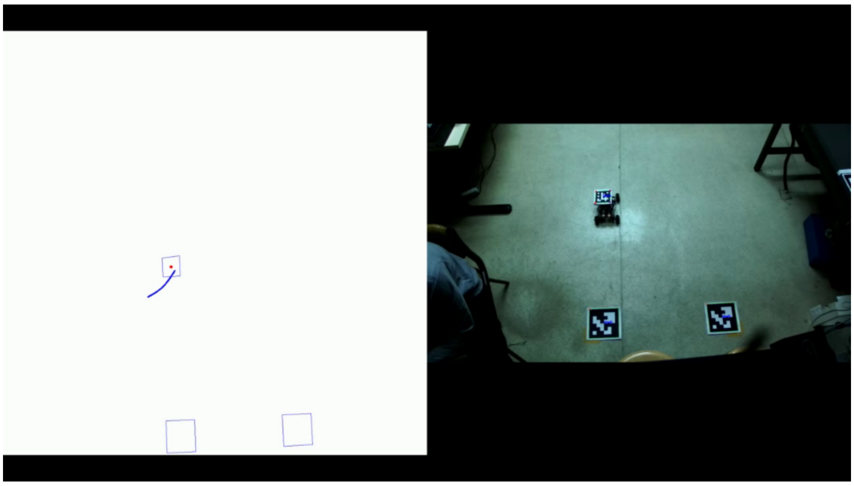

Figure 4.70: Scenario17-a

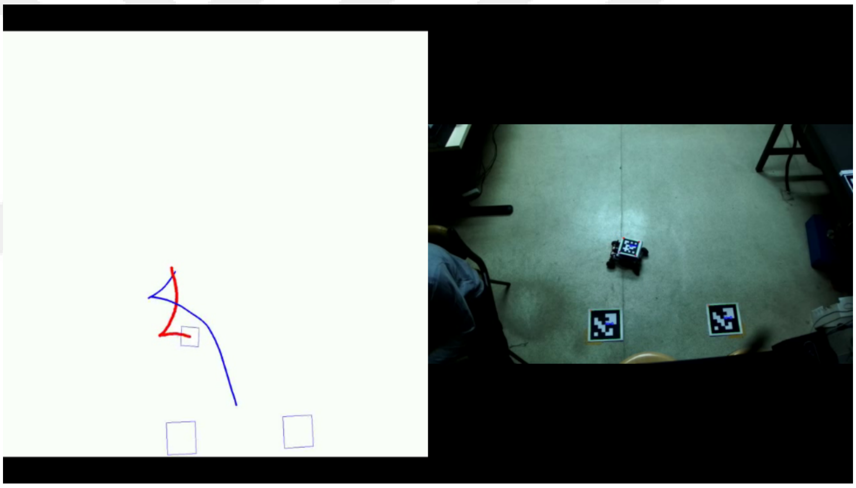

Figure 4.71: Scenario17-b

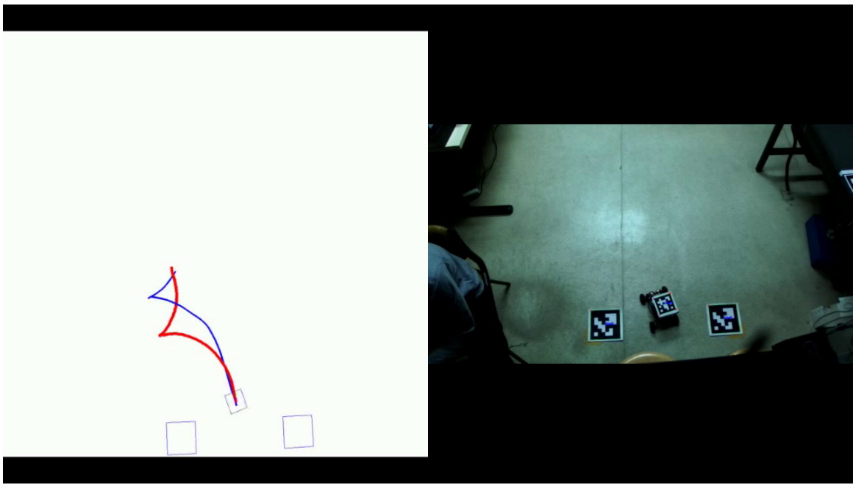

Figure 4.72: Scenario17-c

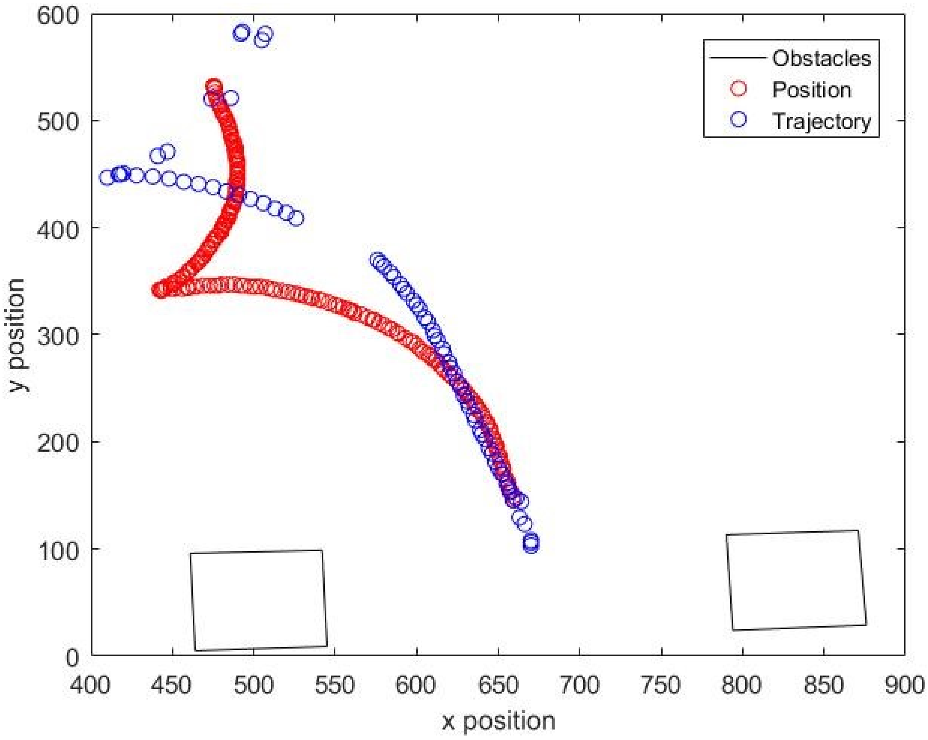

Figure 4.73: Scenario17-d

4.2.18 Scenario 18

In this scenario, the vehicle's start point gets even closer vertically and is on the right side. The vehicle's direction is to the left. The vehicle re-routes a few times in this scenario.

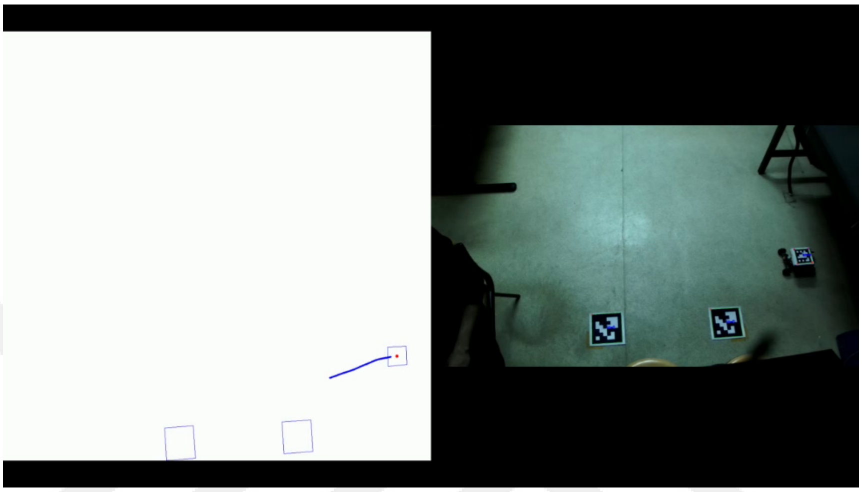

Figure 4.74: Scenario18-a

Figure 4.75: Scenario18-b

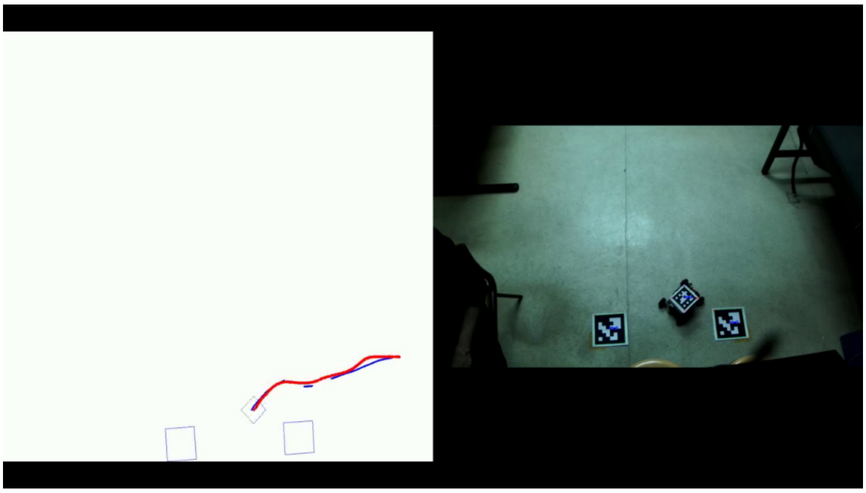

Figure 4.76: Scenario18-c

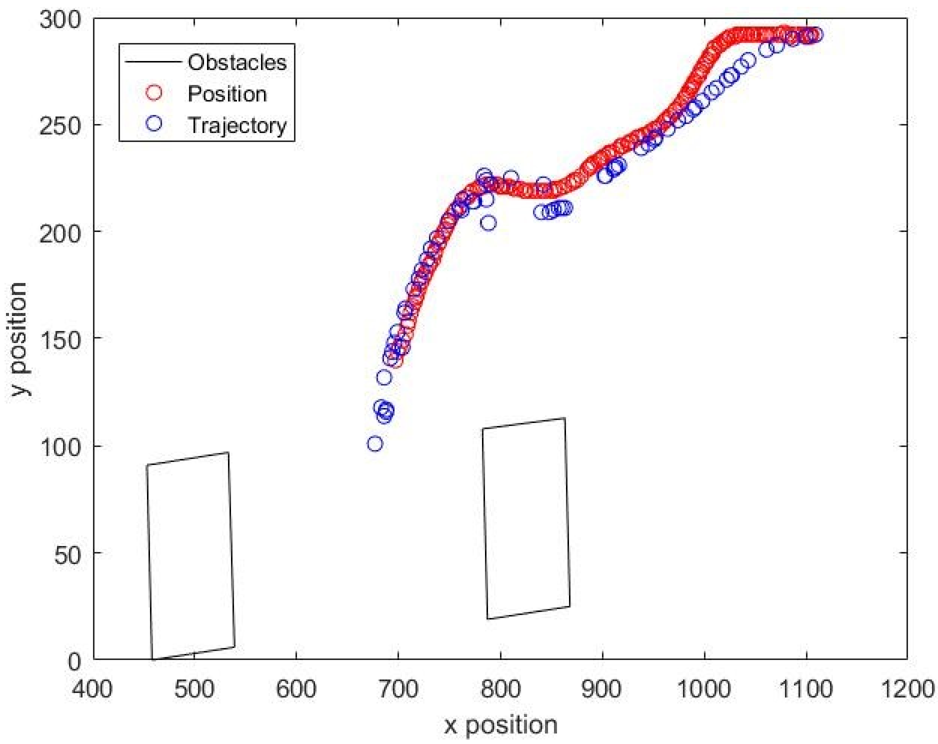

Figure 4.77: Scenario18-d

4.2.19 Scenario 19

In this scenario the vehicle stars from close range, on the left side and facing right. The vehicle re-routes once in the beginning of its journey.

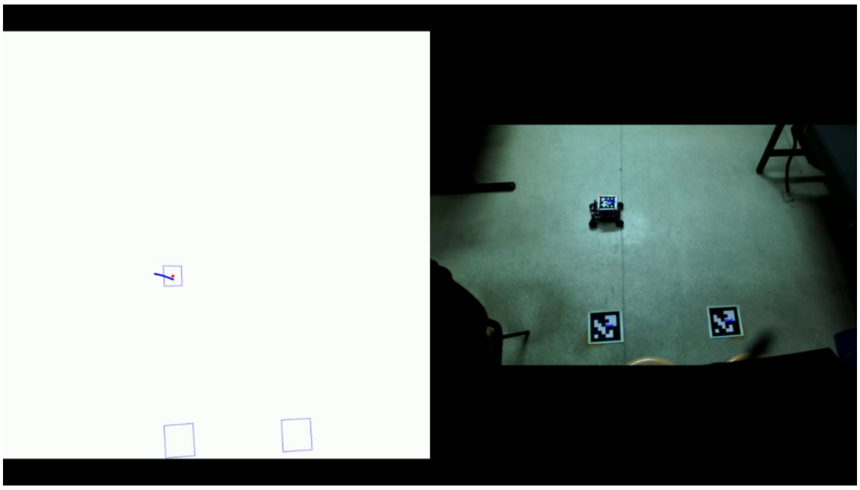

Figure 4.78: Scenario19-a

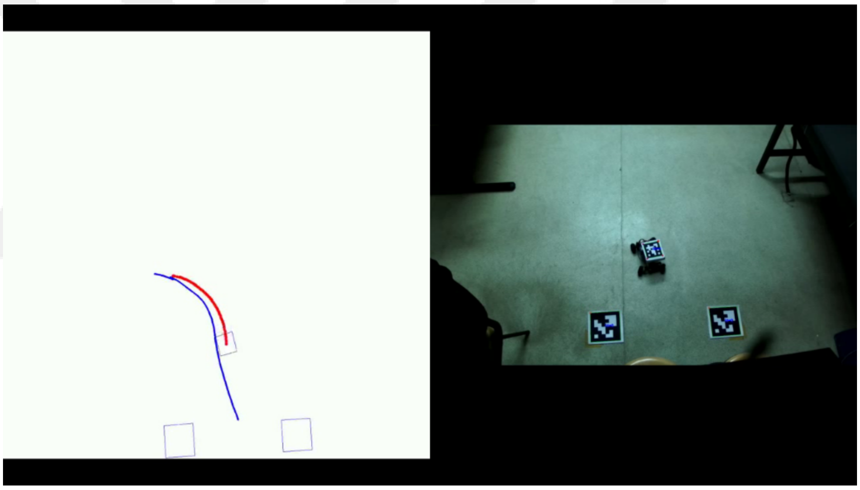

Figure 4.79: Scenario19-b

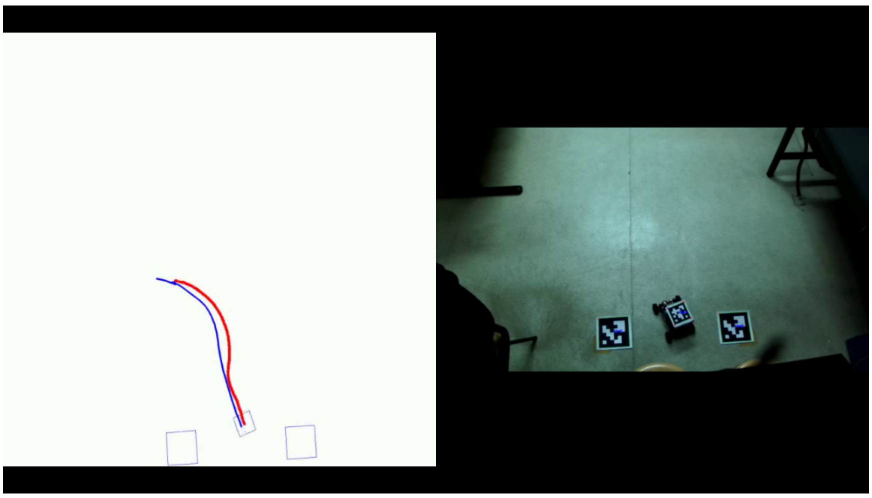

Figure 4.80: Scenario19-c

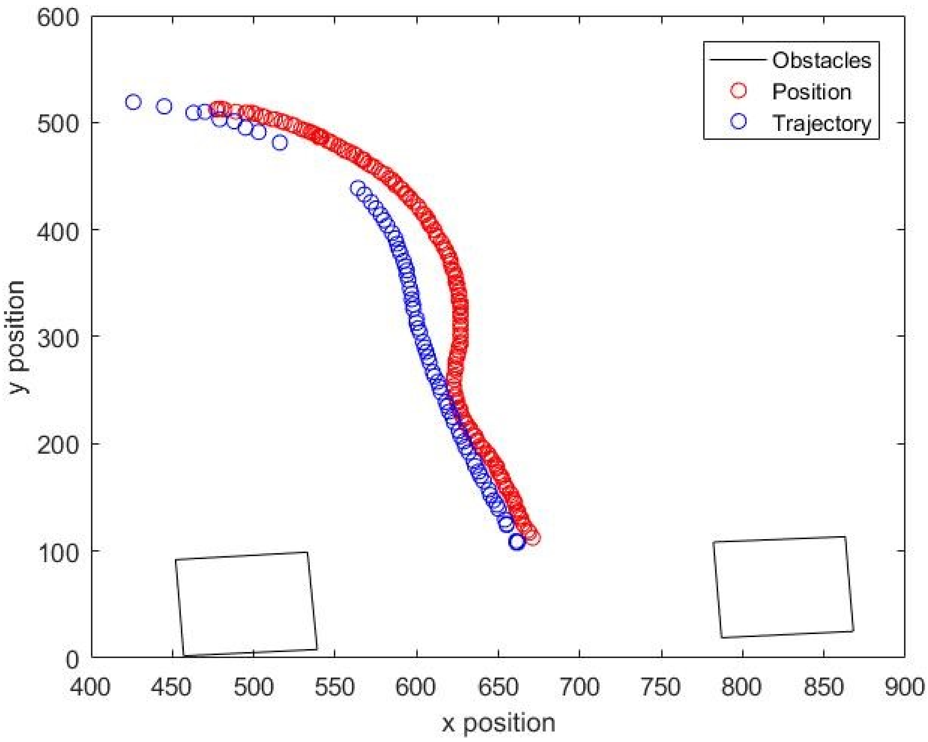

Figure 4.81: Scenario19-d

5 CONCLUSION

In this thesis, a novel real-time path planning algorithm has been presented. Our proposed algorithm is a strong alternative to existing algorithms. Its main strengths are working real-time and being realistic. The experimentation was made with a realistic small-scale electric car and engines on our small car is similar to real life electric engines. Hence, our algorithm is scalable and can work at the same efficiency when applied to full-size electric cars.

The experimental study shows that the algorithm can provide a smooth, feasible path between the starting end goal points. It can also re-route if the initial path cannot reach the goal point for some reason or if the vehicle has drifted away from the path.

REFERENCES

Audi, I. (2010). Reaffirmed Mission for Autonomous Audi TTS Pikes Peak.
URL: *https://web.archive.org/web/20120710202052/http://www.audiusa.com/us/brand/en/tools/news/pool/2010/07/new_look__reaffirmed.html*

Bishop, R. (2005). ALVINN: an autonomous land vehicle in a neural network, *Advances in Neural Information Processing Systems* : 46-55

Blaszczyk, R.L. (2012). The Future is Here: Norman Bel Geddes and the Theater of Time, *Norman Bel Geddes Designs America*, Harry N. Abrams, pp. 76-79.

Carfrae, J. (2011). An automated adventure at the wheel of a driverless BMW.
URL: *https://www.thenational.ae/business/an-automated-adventure-at-the-wheel-of-a-driverless-bmw-1.371963*

Dijkstra, E.W. (1959). A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik 1* : 269 - 271

Doxygen. (2015). Detection of ArUco Markers
URL: *https://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html*

Dudley, D. (2015). Google Self-Driving Car Perfect for Elderly Drivers.
URL: *http://www.aarp.org/home-family/personal-technology/info-2014/google-self-driving-car.html*

Fakoor, M., Kosari, A., Jafarzadeh, M. (2015). Revision on fuzzy artificial potential field for humanoid robot path planning in unknown environment, *International Journal of Advanced Mechatronic Systems Vol. 6* : 174 - 183

Filippetti, J. (2011). EN V electric networked car concept by GM begins pilot testing.
URL: *https://www.designboom.com/technology/en-v-electric-networked-car-concept-by-gm-begins-pilot-testing*

Fitchard, K. (2012). Ford is ready for the autonomous car. Are drivers?.
URL:*https://gigaom.com/2012/04/09/ford-is-ready-for-the-autonomous-car-are-drivers*

Hart, P.E., Nilsson, N.J., Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2* : 100 – 107

Hicks, N. (2017). Nebraska tested driverless car technology 60 years ago.
URL: *https://journalstar.com/news/local/govt-and-politics/nebraska-tested-driverless-car-technology-years-ago/article_a702fab9-cac3-5a6e-a95c-9b597fdab078.html*

Kavralu, L.E., Svestka P., Latombe, J., Overmars, M.H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation Vol. 12, No.4* : 566 - 580

King-Hele D. (2002). Erasmus Darwin's Improved Design for Steering Carriages--And Cars, *Notes and Records of the Royal Society of London Vol. 56* : 41-62

Lavalle, S.M. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning, Iowa State University.

Lego. (2009). LegoMindstorms NXT 2.0.
URL: *https://www.bricklink.com/v2/catalog/catalogitem.page?S=8547-1#T=S*

Lego. (2009). 10287 Intelligent NXT Brick.
URL: https://lego.fandom.com/wiki/10287_Intelligent_NXT_Brick

Leohold, J. (2011). Driving without a Driver – Volkswagen presents the "Temporary Auto Pilot, *Volkswagen* : 5 - 11

Ozguner, U., Acarman, T., Redmill, K. (2011). Autonomous Ground Vehicles, Artech House

Pomerleau, D. A. (1989). ALVINN: An Autonomous Land Vehicle in a Neural Network, *Advances in Neural Information Processing Systems 1*, Morgan-Kaufmann, pp. 305-313.

Richards, N.D., Sharma, M., Ward, D.G. (2004). Hybrid A*/Automaton Approach to On-Line Path Planning with Obstacle Avoidance, *AIAA 1st Intelligent Systems Technical Conference 1* : 1 - 17

Sentinel, T.M. (1926). 'Phantom Auto' Will Tour City.

URL: *https://news.google.com/newspapers?id=unBQAAAAIBAJ&sjid=QQ8EAAAAIBAJ&pg=7304,3766749*

Sins, J. (2013). Toyota sneak previews self-drive car ahead of tech show.
URL: *https://www.bbc.com/news/technology-20910769*

Thrun, S. (2010). Toward robotic cars, *Communications of the ACM* : 99 - 106

Vanderbilt, T. (2012). Autonomous Cars Through the Ages.
URL: *https://www.wired.com/2012/02/autonomous-vehicle-history*

Wik, I. (2012). Futurama (New York World's Fair).
URL: *https://en.wikipedia.org/wiki/Futurama_(New_York_World%27s_Fair)*

Zed. (2019). ZED Stereo Camera.
URL: *https://www.stereolabs.com/zed/*

BIOGRAPHICAL SKETCH

Ahmet Emre Danışman was born on July 9th, 1993. He was graduated from Kabataş Erkek High School in 2012. Later, he got his bachelor's degree from Bilkent University Computer Science Department in 2016. In 2019, he got his master's degree from Galatasaray University Computer Engineering Department. The master's thesis is this document and he was supervised by Prof. Dr. Tankut Acarman.

So far, he has two academic publications. "Human Activity Recognition With Mobile Phone Sensors: Impact of Sensors and Window Size" was published at Signal Processing and Communications Applications Conference (SIU) in 2018. The second publication was a representation of his master's thesis study, "A Car-Like Robotic Experimentation for Path Planning Study" was presented at 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications and published at SETSCI in 2019.

His top working topics are deep learning and autonomous vehicles.

PUBLICATIONS

Danışman, A.E., Acarman, T. (2019) A Car-Like Robotic Experimentation for Path Planning Study, *SETSCI, Vol.4(5)* : 55 - 60