

# A cost-effective GPS-aided autonomous guided vehicle for global path planning

Gorgees S. Akhshirsh<sup>1</sup>, Nawzad K. Al-Salihi<sup>2</sup>, Oussama H. Hamid<sup>3</sup>

<sup>1,2</sup>Department of Computer Science and Engineering, University of Kurdistan Hewar (UKH), Erbil, Kurdistan Region, Iraq

<sup>3</sup>University of Nottingham, United Kingdom (UK)

## Article Info

### Article history:

Received Aug 17, 2020

Revised Oct 13, 2020

Accepted Dec 5, 2020

### Keywords:

Autonomous guided vehicles (AGVs)

GPS

Path planning

Deliberative and reactive robot architecture

## ABSTRACT

This paper presents a robotic platform of a cost-effective GPS-aided autonomous guided vehicle (AGV) for global path planning. The platform is made of a mechanical radio controlled (RC) rover and an Arduino Uno microcontroller. An installed magnetic digital compass helps determine the right direction of the RC rover by continuously synchronising the heading and bearing of the vehicle. To ensure effective monitoring of the vehicle's position as well as track the corresponding path, an LCD keypad shield was, further, used. The contribution of the work is that the designed GPS-aided AGV can successfully navigate its way towards a destination point in an obstacle-free outdoor environment by solely relying on its calculation of the shortest path and utilising the corresponding GPS data. This result is achieved with a minimum error possible that lies within a circle of one meter radius around the given destination, allowing the devised GPS-aided AGV to be used in a variety of applications such as landmine detection and removal.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Oussama H. Hamid,  
University of Nottingham,  
United Kingdom (UK)  
Email: oh.hamid@gmail.com

## 1. INTRODUCTION

Automated systems are an inevitable futuristic vision of human societal development [1–3]. A vital determinant of any innovative infrastructure in such a vision [4] is the incorporation of autonomous guided vehicles (AGVs). An autonomous guided vehicle (AGV) is a self-steering mobile robot which strives to achieve its goals by maintaining an independently chosen course while navigating a complex and uncertain environment. Rather than acting upon constant human instructions, an AGV system plans proactively its course of movement between two points by taking advantage of available data. Such data could have been gained from previously experienced contexts or stems from currently incoming sensory streams. This subtask of the robot is called 'path planning'.

There are two architectures of path planning: reactive and deliberative [5, 6]. This paper presents a deliberative mobile robot that belongs to the category of global planners. Our main contribution in this work lies in the design and implementation of a cost-effective, integrated robotic system which can successfully navigate its way towards a destination point by solely relying on its calculations of the shortest path and utilising the corresponding GPS data. The implemented GPS-aided robotic system comprises several electronic devices; namely, an Arduino microcontroller, a GPS, and a magnetic compass. The mobile robot finds the path between two points in any open area and obstacle free environment (please see the supplemented video material). Im-

portantly, the designed guiding mechanism and the deployment of the Longitude and Latitude parameters of the destination point via Google Maps, enable the devised robot to achieve correct motion planning with high accuracy. Specifically, the range of error lies within a circle of one meter radius around the destination point. Furthermore, in case of intending to visit more than one destination at a time, the robot neither starts from scratch nor it initialises the search for a new path once again. Rather, the robot orders the destinations with respect to their coordinates and handles these destinations as waypoints in the provided path, visiting them sequentially as it moves along the path to the final destination point. This is particularly efficient when having a far destination point, and, hence, a relatively long path, for it would reduce the resultant error cumulatively.

## 2. RESEARCH METHOD

### 2.1. Background study

Mobile robots are main testbed platforms of the current industrial revolution (Industry 4.0) [7]. For robots to become able to move, several problems should be solved. Path planning is considered a fundamental problem and is one of the most studied problems in robotics [8–14]. Research distinguishes between two architectures of path planning: reactive and deliberative path planning [5, 6]. In the reactive paradigm, a low-level control is implemented that produces an autonomous behaviour through a loop of sense-act couplings. In contrast, an AGV system which operates in a deliberative fashion applies a top-down approach that enables the robot to first sense relevant components of the operating environment, then plan its next step, and finally proceed with the corresponding action [6, 15].

Other researchers refer to the reactive and deliberative planning architectures as local and global planners, respectively [9, 16–19]. Local path planning assumes no a priori knowledge about the environment. Instead, the robot perceives existent environmental obstacles through the embedded sensors and generates its path in real time with relatively low computational effort [20]. By contrast, global path planning utilises a previously known map of the environment with all what it includes to generate a relatively safe path. Consequently, algorithms within the class of global path planning converge for sure only in static environments. Dynamic environments, however, require re-planning, which results in increased computational times for the applied method.

### 2.2. Setup and hardware components

Different types of sensors and control architectures can be used in order to build automated systems [21–23]. In the present work, a toy radio controlled (RC) car with two DC motors was used: one for the front tires and another for the rear tires with dimensions of approximately 25 cm long, 11 cm wide, and 11.5 cm high so that all the devices will fit on top of it plus the portable batteries to charge the devices along the ride. Setup and hardware components as shown in Figure 1.

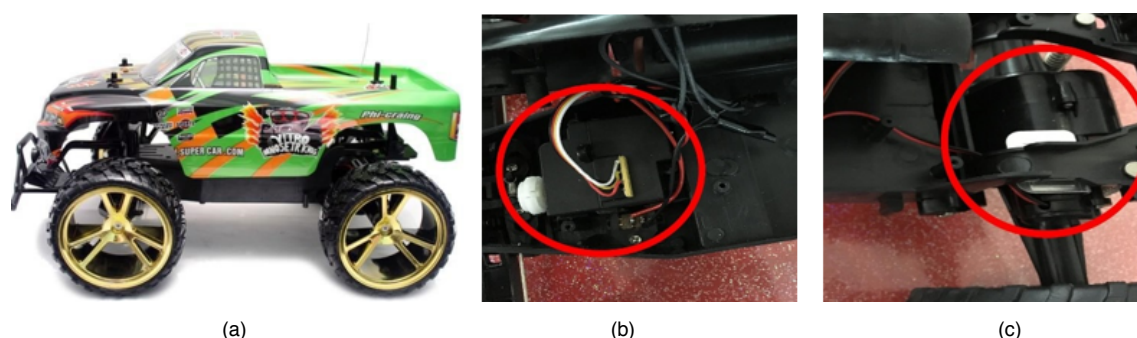


Figure 1. Setup and hardware components: (a) RC Car, (b) Front DC Motor, (c) Rear DC Motor

The front DC motor in this car is a 9 Volt electric circuit which requires at least 3 Ampere to run. It's function is basically to rotate the front tires either 90 degrees to the left or 90 degrees to the right. We could have used a servo motor. However, we opted for the DC motor to avoid programming complexity besides cost issues. Importantly, the resulting difficulty in specifying the turning angle of the tires when using the DC motor is engineered through an algorithmic solution. Such a solution will make the vehicle rotate in a smaller

diameter enhancing accuracy as would be demonstrated later in the upcoming section. The other DC motor powers the rear tires of the RC car and has the same technical specifications as the front one. The motors are fed by relays that work as a controller which takes its input by an Arduino. Since the relays on the original RC car were fixed and unadaptable to be plugged with an Arduino, they were replaced by two SainSmart 2-Channel relays to control the front and rear motor.

An H-bridge circuit is sorted as the electronic circuit that empowers a voltage to be connected over a heap in either heading. It is very common in the robotic industry especially for driving DC motors either backward or forward. This circuit allows current flowing through it in both directions, and the H-bridge current flow is controlled using the Pulse Width Modulation (PWM) protocol. This method is basically used to generate an analog signal from a digital device. Figure 2(a) shows the assembled RC car with all previously described digital and electronic components.

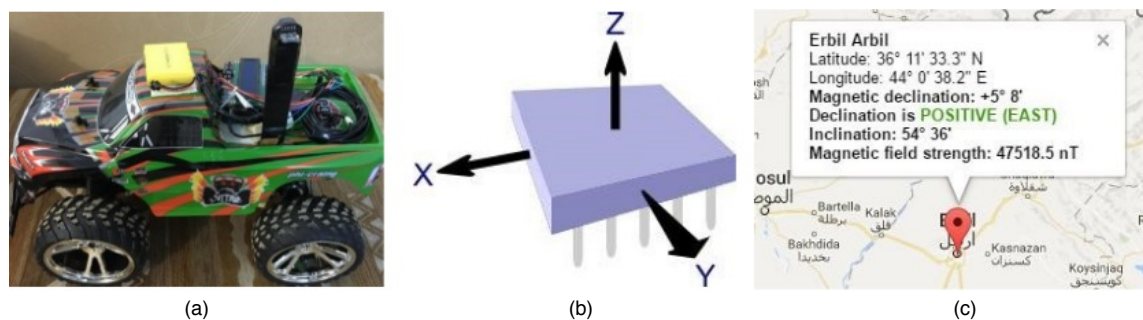


Figure 2. Obtaining the robot's current heading: (a) Assembled mobile robot, (b) Compass axes, (c) Erbil declination point

### 2.3. The algorithm

For the robot to autonomously move to a specific location, it requires that at any point the GPS location longitude and latitude can be taken from an online map (e.g., Google Map) and given to the robot. The robot makes its calculations and moves towards its goal on the basis of the given points' coordinates. Furthermore, there are many other approaches and attempts applied for achieving an autonomous guided GPS robot project as discussed in the section of background study. Our methodology, however, consists of a combination and contribution of several electronic devices to provide the needed data in order to guide the rover precisely to the destination. Figure 3 shows the algorithmic steps.

### 2.4. Obtaining the robot's current heading

Figure 2(b) shows the 3-Axis (X, Y, and Z) digital compass (HMC5883L data sheet), which was used to find the heading angle with respect to the magnetic north. The resulting value of the heading angle varies from 0 (which is north) to 360 degrees. This helps overcoming the tilting issue. To explain, in case the compass is held in a leaning degree of up to 45 degrees to right or left, having only X and Y magnitudes will result in inaccurate readings (because the X and Y magnitudes are not aligned with the axes of earth's magnetic field). Hence, a third axis, the Z axis, solves the magnetometer problem of the compass leaning even up to 45 degrees. From the numerals of axes X and Y we can find the current heading of the robot in degrees (from 0 to 360 degrees) by calculating the vector defined within values of Y and X and this is performed from the formula:

$$\text{Heading} = \arctan2(y, x) \quad (1)$$

This radians-based value is converted into degrees as follows:

$$\text{Heading} = (180 \times \arctan2(y, x)) / \pi \quad (2)$$

with  $\pi \approx 3.14$ . In case of obtaining a negative result we add 360. Moreover, to eliminate the difference between magnetic and geographic (true) Norths [24], we add the value of the magnetic declination in Erbil, our experimental place, which is 5.8 degrees (Figure 2(c)). Furthermore, we need to change this degree value into

radians by applying the formula:

$$\text{angle in radius} = (\text{angle in degrees} \times \pi) / 180 \quad (3)$$

then add it to the first Heading formula in radians before changing it into degrees. As a result, the final Heading becomes:

$$\text{Heading} = (\arctan2(y, x) + \phi) \times 180 / \pi \quad (4)$$

where  $\phi$  is the magnetic declination angle in radian. Using equations 1 to 4 the X, Y, Z readings which were provided by the compass (10 readings each second) are transferred to heading in degrees from 0 to 360.

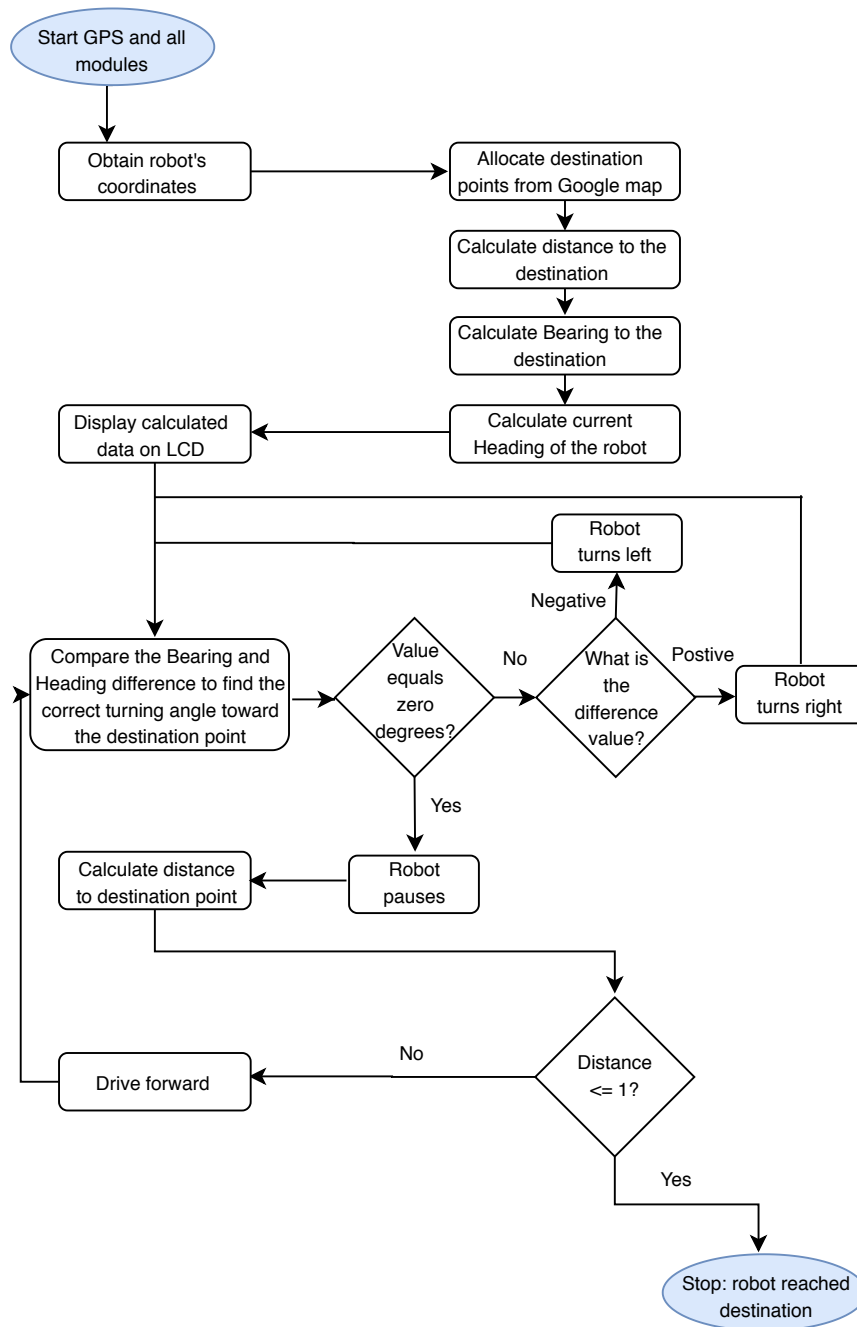


Figure 3. Flow chart of the autonomous system

### 3. RESULTS AND DISCUSSION

#### 3.1. Real-time experiments

To test the vehicle's performance in real time, we set six points (Table 1) on the map by including their coordinates in the sketched code manually (more points could have been taken) and the vehicle will move autonomously toward them in sequence. Figure 4(a) shows the points marked a rectangular shape on the map with longitude values shown against corresponding latitude values in Figure 4(b). Each point is approximately 10 meters away from the other on the map. A home point representing the robot's initial position is marked as point A. Other points with lexicographical labels B, C, D, E, and F are to be visited by the robot. Our tests were done within the university garage at the University of Kurdistan Hewlêr (UKH) in Erbil.

Table 1. Waypoints coordinates

Point Label	Latitude	Longitude
A	36.181453	44.006763
B	36.181552	44.006755
C	36.181510	44.006690
D	36.181446	44.006786
E	36.181373	44.006835
F	36.181400	44.006885



Figure 4. Real-time experiments: (a) Marked points on the map, (b) Longitude against latitude

Once the RC car is placed at the initial point (A), pointing towards waypoint (B), it starts moving within approximately one second. As a result, the GPS to observe its current coordinates and plots the longitude and latitude on the LCD keypad. Importantly, this one second delay only occurs when initializing the receiver of the GPS shield signal at the beginning. Moreover, longitude and latitude values can also be viewed on the serial of Arduino IDE when it is connected to the computer through USB cable.

As per Table 1, the calculation of bearing and distance toward the first waypoint destination (B) will take the coordinates of initial and first waypoint. But when calculating the distance and bearing for the second waypoint destination, we take the coordinates of the initial and first waypoint, as the robot assumes we are at the home waypoint when calculating toward the first point. To check the accuracy of the calculation in the sketched code for both distance and bearing, we will take the route between A and B coordinates as an example and compare the results to a trusted online website for calculating distance and bearing between two given points called Sun Earth Tool website. Initial point to the first waypoint put into test in the website tool and the following results were obtained in Figure 5(b). Compared with the readings from the GPS we got values on the serial port as shown in Figure 6(a). When reaching waypoint one, the car will drive to waypoint two and do the same process of calculation taking the reached point coordinates at waypoint one as a reference toward the second waypoint and so on until reaching the last destination. The calculated values by the program as shown in the figure above are very close to the calculation of the website tool within an error range of no more than a meter distance and small error degrees regarding bearing. These calculated digits will be displayed on the LCD keypad as shown in Figure 6(b). As presented in the figure above, the distance will be exhibited on the first row the 7th character on the LCD, and Bearing value on the second row first character on the left side



of the LCD. The first row on the first character the turning direction of the robot is displayed as seen on the LCD and the heading degree is shown on the second row 8th character, plus exhibiting the waypoint number that the robot is heading toward it on the top right hand first row 13th character. Just like the GPS coordinates, the bearing and distance values keep updating and are plotted on the keypad in each loop (approximately 3 seconds) based on the new received data from the GPS shield while the rover is moving. In addition to the coordinates, bearing and distance another vital statistic is required to guide the vehicle accurately towards the destination point which is the heading of the RC car. The digital compass on the vehicle will calculate the current heading of the car showing the direction of the vehicle and display it on the keypad as shown in Figure 6(b). The heading calculated by the compass while the car is moving and rotating keeps updating by providing around 10 readings each second.

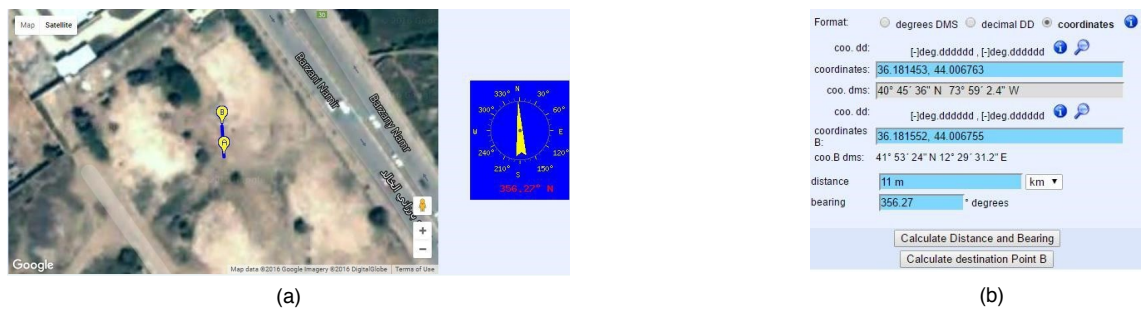


Figure 5. Accuracy of route calculation between initial and first points: (a) Initial and first points on the map, (b) Distance and bearing from initial toward first waypoint on the website

### 3.2. Testing results of the autonomous system

As demonstrated in Figure 6(c), the driving accuracy of the robot was considerably high at most of the attempts. The AGV directed itself accurately toward the waypoints within a maximum deviation of approximately 2 meters to right and left from the straight path between the waypoints. Error rates correspond to the discrepancy between the coordinate values of the waypoints that mark the rectangular shape in Figure 4 and those determined by the robot's path over repeated testing of the robot's performance.

In some of the trials the rover ran into a deadlock, as it overdrove or overturned the aimed waypoint. This caused the rover to move around the same point for 360 degrees. As an average of all the tests, it took the rover less than 10 minutes to drive throughout the 6 waypoints and reached an hour back to the home point at the last destination. The error range recorded was within 1 meter circle around the destination point. The batteries lasted for the entire time of testing (almost 2 hours).

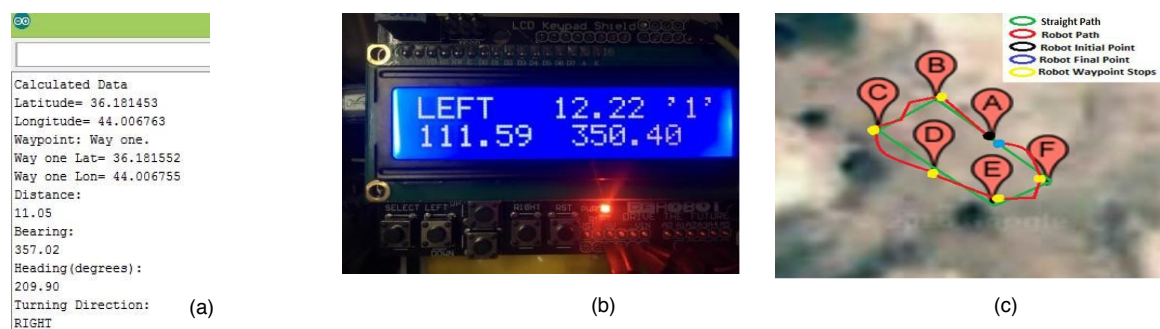


Figure 6. Testing results: (a) Obtained calculation from the robot's initial point (waypoint A) toward first point (waypoint B), (b) LCD keypad display showing calculated data, (c) Robot path error rates

#### 4. CONCLUSION

In this work we demonstrated how to build a functional autonomous guided vehicle (AGV) with low budget based on a GPS receiver and a digital compass. Despite the variety of industrial applications AGVs can be used in, our work was motivated by the idea of developing autonomous robots that could be deployed in the removal of landmines in Kurdistan Region, the place where this work was achieved. The high performance and cost-effectiveness of our AGV was constructed by using an Arduino microcontroller [25], GPS and installing a digital compass on a simple modified radio control car chassis. Not only did we design a regular rover driving. But we also built a prototype of a smart self-guiding car that can calculate the shortest path and by using the heading degree it drives its way through pre-programmed waypoints on the map. With such a prototype of our GPS-aided AGV we could achieve a considerably high driving accuracy at most of the tests. The error range recorded was within one meter circle around the destination point. Importantly, our design provides other researchers further possibilities to improve performance and accuracy of the presented GPS-aided AGV by considering contextual changes so as to help switch between paths [26, 27], or by applying certain filtering techniques on the GPS signal received such as Kalman filtering or Real Time Kinematic-Differential Global Positioning System (RTK-DGPS) [28].

#### ACKNOWLEDGEMENT

The authors gratefully acknowledge the use of service and facilities of the University of Kurdistan Hewlêr (UKH) and thank Dr. Wrya Monnet for technical support as well as the editor and anonymous reviewers of the Bulletin of Electrical Engineering and Informatics for helpful comments and suggestions.

#### REFERENCES

- [1] S. Mazlan, *et al.*, "Kinematic variables for upper limb rehabilitation robot and correlations with clinical scales: A review," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 75–82, 2020.
- [2] N. L. Smith, J. Teerawanit, O. H. Hamid, "AI-Driven Automation in a Human-Centered Cyber World," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 3255–3260.
- [3] O. H. Hamid, N. L. Smith, and A. Barzanji, "Automation, per se, is not job elimination: How artificial intelligence forwards cooperative human-machine coexistence," In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. IEEE, 2017, pp. 899–904.
- [4] F. Dressler, P. Handle, and C. Sommer, "Towards a vehicular cloud-using parked vehicles as a temporary network and storage infrastructure," in *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities*. ACM, 2014, pp. 11–18.
- [5] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [6] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn, "Path planning and obstacle avoidance for autonomous mobile robots: A review," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, 2006, pp. 537–544.
- [7] M. A. Muslim, *et al.*, "Development of a quadruped mobile robot and its movement system using geometric-based inverse kinematics," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1224–1231, 2019.
- [8] A. Dragan, and S. Srinivasa, "Integrating human observer inferences into robot motion planning," *Autonomous Robots*, vol. 37, no. 4, pp. 351–368, 2014.
- [9] M. Elbanhawi, and M. Simic, "Sampling-based robot motion planning: A review," *Ieee access*, vol. 2, pp. 56–77, 2014.
- [10] A. H. Karami, and M. Hasanzadeh, "An adaptive genetic algorithm for robot motion planning in 2D complex environments," *Computers & Electrical Engineering*, vol. 43, pp. 317–329, 2015.
- [11] S. H. Klidbary, S. B. Shouraki, and S. S. Kourabbaslou, "Path planning of modular robots on various terrains using Q-learning versus optimization algorithms," *Intelligent Service Robotics*, vol. 10, no. 2, pp. 121–136, 2017.
- [12] J.-C. Latombe, *Robot motion planning*, Springer Science & Business Media, 2012, vol. 124.
- [13] S. Murray, W. Floyd-Jones, Y. Qi, D. J. Sorin, and G. D. Konidaris, "Robot Motion Planning on a Chip," in *Robotics: Science and Systems*, 2016.
- [14] K. L. Su, H.-S. Su, S.-W. Shiao, and J.-H. Guo, "Motion planning for a landmine-detection robot," *Artificial Life and Robotics*, vol. 16, no. 3, pp. 277–280, 2011.
- [15] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [16] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-based methods for motion planning with constraints," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 159–185, 2018.
- [17] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An artificial potential field based mobile robot navigation method to

- prevent from deadlock,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 5, no. 3, pp. 189–203, 2015.
- [18] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [19] W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and decision-making for autonomous vehicles,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.
- [20] F. Martinez, E. Jacinto, and F. Martínez, “Using bacterial interaction and stereoscopic images for the location of obstacles on autonomous robots,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 906–913, 2020.
- [21] S. E. Li, et al., “Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017.
- [22] Ö. Ş Taş, et al., “Making bertha cooperate-team annieway’s entry to the 2016 grand cooperative driving challenge,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1262–1276, 2018.
- [23] C. Urmson, et al., “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [24] A. C. Sparavigna, “Magnetic Compasses and Chinese Architectures,” arXiv preprint arXiv:1709.07056, 2017.
- [25] N. L. Mohammed, M. S. Aziz, and O. I. AlSaif, “Design and implementation of robot control system for multistory buildings,” *Telkomnika*, vol. 18, no. 5, pp. 2682–2689, 2020.
- [26] O. H. Hamid, “The role of temporal statistics in the transfer of experience in context-dependent reinforcement learning,” in *2014 14th International Conference on Hybrid Intelligent Systems*. IEEE, 2014, pp. 123–128.
- [27] O. H. Hamid, “A model-based Markovian context-dependent reinforcement learning approach for neurobiologically plausible transfer of experience,” *International Journal of Hybrid Intelligent Systems*, 12(2), pp.119–129.
- [28] N. K. Al-Salihi, “Precise positioning in real-time using GPS-RTK signal for visually impaired people navigation system,” Ph.D. dissertation, *Brunel University School of Engineering and Design PhD Theses*, 2010.

## BIOGRAPHIES OF AUTHORS



**Gorgees S. Akhshirsh** is a Monitoring and Evaluation Officer at Unopnteper in Erbil, Kurdistan Region - Iraq. Having completed his Bachelor studies in Computer Engineering in October 2014, he earned in December 2016 a Master’s degree in Computer System Engineering from the University of Kurdistan Hewlêr (UKH). His research interests focus on the design and implementation of intelligent solutions for information and communication technologies, in particular, in the fields of autonomous guided vehicles and path planning.



**Nawzad K. Al-Salihi** is an Assistant Professor at the Department of Computer Science and Engineering (CSE) at University of Kurdistan Hewler (UKH). Dr. Al-Salihi has an extensive academic background, which includes a Ph.D. in Electronic and Computer Engineering from Brunel University, London, UK (2010), Thesis title: “Precise Positioning in Real-Time using GPS-RTK Signal for Visually Impaired People NavigationSystem”; MSc in Advanced Manufacturing Systems from Brunel University, London, UK; BSc in Automation Engineering from University of Skovde, Sweden. Dr. Al-Salihi has widespread professional experience in satellite navigation systems (GPS, GLONASS and GALILEO), wireless mobile communications and advanced manufacturing systems. He has published several academic journal articles and presented at several international conferences.



**Oussama H. Hamid** is an Assistant Professor in Computer Sciences. Professionally trained and educated in Germany, he earned in 2011 a Doctorate of Natural Sciences (Dr. rer. nat.) from Otto-von-Guericke University Magdeburg. Back in 2002, He was awarded a Master’s degree as Graduate Engineer (Dipl.-Ing.) in Computational Visualistics and two German Vordiplom Certificates (≈BSc) in Computer Science and Mathematics in 2000 and 1998, respectively. Since 2003, Dr. Hamid has been investigating both theoretical and practical aspects of human-centred Artificial Intelligence, Neural Network Functions, Cognitive Systems and Big Data Analysis. His research findings have been published in several European and international refereed journals and conferences. To his scholarly responsibilities, Dr. Hamid is an Accredited Researcher at the Office for National Statistics (ONS) in London (UK), a member of the IEEE Systems, Man, and Cybernetics (SMC) Technical Committee on Soft Computing, and a member of the European Institute for Systems and Technologies, Control and Communications (INSTICC).