Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 1821–1830

Procedia
Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Application of Paraconsistent Annotated Evidential Logic Eτ for a Terrestrial Mobile Robot to Avoid Obstacles

Flavio Amadeu Bernardini[a*], Marcia Terra da Silva[a], Jair Minoro Abe[a]

[a] *Graduate Program in Production Engineering - Paulista University. Rua Dr. Bacelar 1212*

Abstract

One of the significant challenges in contemporary robotics concerns navigation systems and autonomous robots' location, being the detection and avoidance of obstacles an essential part of this system. This article proposed Paraconsistent Annotated Evidential Logic Eτ in the control algorithm for a terrestrial robot, and thus contribute with navigation systems, to avoid possible collisions with obstacles. A prototype terrestrial robot was then built with ultrasonic sensors for obstacle detection and a servomotor in steering control. Several performance tests were performed. It was noticed that by shortening the perception distances of the sensors, the performance of the robot's displacement proved to be satisfactory.

Keywords: Paraconsistent annotated logic, Servomotor, Terrestrial mobile robot, Robotics

1. Introduction

Nowadays, many researchers on robotics stand out in the development of navigation techniques and the most diverse practical applications of the use of robots in the activities of the contemporary world. The authors [1] highlight and analyse the most widespread navigation techniques currently applied in exhaustive research. This study includes navigation of crewless vehicles and 2D ground robots and aerial and aquatic vehicles in 3D navigation. According to the authors, Wheel Odometry (WO) is one of the simplest forms of localisation that has been used in many two-and four-wheel robots. Roversare prominent examples of these robotic vehicles used in the exploration of Mars. The WO method is based on wheel encoders mounted on a robot to track the number of

---

[*] *Corresponding author email address: flavioambernar@gmail.com*

10.1016/j.procs.2021.08.187

revolutions each wheel has made. The number of processes is integrated into a dynamic modeller to determine the robot's current position relative to the starting point [2]. The Inertial Navigation System (INS) works based on Inertial Measurement Unit (IMU) sensors, consisting of an accelerometer that measures non-gravitational acceleration and a gyroscope that measures orientation based on the measurement of gravity and magnetism. These sensors' small size and low energy consumption have made them ideal for navigation resource systems for drones and micro-robots. Also, such systems do not require an external reference to estimate the position of a platform accurately. However, these systems suffer from a drift problem due to errors originating from different sources, such as constant errors in gyroscope and accelerometers measurements. Therefore, the INS is not used for applications that require localisation for long periods [3].

The authors [4] have developed a system for recognising artificial landmarks used in the location of interior robots. To this end, they introduced a robust and straightforward reference design to propose a reliable landmark recognition approach capable of handling the influence of ambient lighting conditions and the incompleteness of landmarks. These artificial landmarks were fixed on the hallway ceiling of an office environment, a Logitech H.D. Pro C910 regular-view webcam fixed on the Pioneer 3-DX 1 mobile robot and facing the ceiling. The webcam extracts images from these landmarks, and so an Intel Core 2 computer, due to 2.0 GHz CPU and 2G RAM processes the images of these landmarks and estimates the position and orientation of the mobile robot.

To solve a navigation problem for autonomous agricultural machines, the authors [5] developed a method through the MATLAB Neural Network toolbox. The authors used a small robot to test the method, in which five SRF05 ultrasonic sensors were used to train the neural network so that it can recognise a cylinder, cone, and parallelogram. Once trained, the algorithm is tested by the autonomous navigation vehicle in an enclosure where the three objects to be recognised were present. Besides, the authors researched that the use of low-cost electronic components in robotics projects does not prevent the possibility of obtaining good results.

The authors [6] developed an integrated system to determine the location of mobile robots in a known environment. For this, first, the images were acquired from an aerial camera located in a fixed position. Thus, the received images were processed by the image processing algorithm. In this way, the angle of position and rotation of the mobile robot was measured. The authors conclude that the camera images and data from the encoder sensors for odometry were combined with the Extended Kalman Filter (EKF) to ensure the correct location of the robot in the previously known environment.

Because of what has been exposed, this work aims to contribute to the development of navigation techniques of terrestrial autonomous robots by using Paraconsistent Annotated Evidential Logic Eτ in decision making to avoid obstacles. Thus, the methodology adopted was based on constructing an autonomous terrestrial mobile robot equipped with ultrasonic sensors to detect possible obstacles. A servo motor was used to control the robot's direction, and then a paraconsistent algorithm was developed for mobile robot navigation.

2. Description of Hardware Sensor, Microcontroller, and Servomotor

Electronic and practical tests were performed on the servomotor and the ultrasonic sensor to verify the performance of the main components responsible for the robot's displacement movements' accuracy. The Arduino model ATmega 2560 microcontroller was chosen for the test since it has the digital inputs and outputs required for all the prototype's peripherals. The microcontroller has a UART module that allows Bluetooth communication with mobile devices, such as mobile phones and tablets. Moreover, being an Open Source platform, the Arduino allows versatility of applications and control of peripherals and has much flexibility in C Language programming actions, which contributed to developing the paraconsistent algorithm used in the project [7].

Most differential unit mobile robot prototypes use two aligned main wheels as traction and a freewheel that supports the other part of the chassis. In these cases, the robot's direction is controlled by the difference in the main wheels' rotation traction, and the freewheel tends to point to the side with the highest rotation. This type of project has a high probability of freewheeling slippage and, consequently, course errors. The main difference of this proposal is in the use of a servomotor in the direct control of the robot, giving more precision to the positioning of the prototype.

To meet the requirements of the prototype, the SG90 model of the Tower Pro of 2.0kgf.cm was chosen, which physical aspect, extracted from the manufacturer's manual, is found in Fig. 1 (a). Fig. 1 (b) shows the waveform of the servo control signal, whose duty cycle varies from 1ms to 2ms so that the servo performs a movement of -90º to +90º about the central position [8].

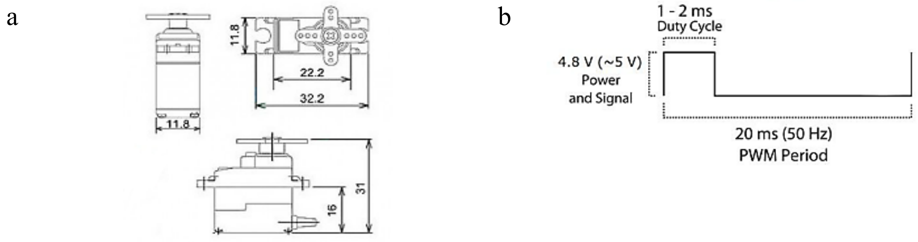

Fig. 1 Servomotor's dimensions (a), Electric servomotor control signal (b)

To verify the performance of the servomotor and ultrasonic sensor chosen for the project, a Tektronics model 1102B oscilloscope was used. Then, technical tests were performed to verify the electrical signals sent by the ATmega 2560 microcontroller. Fig. 2 observes on the oscilloscope screen a signal with the duty cycle of 1.520ms, between the two reading cursors, and a frequency of 49.99Hz. The microcontroller generated this signal so that the servomotor would point to the central position. Compared with the manufacturer's manual Fig. 1 (b), a slight error of 0.020ms is noted in the duty cycle signal and a frequency almost equal to 50Hz. These minimal deviations are expected in practical servomotor applications and do not compromise performance for the desired application.

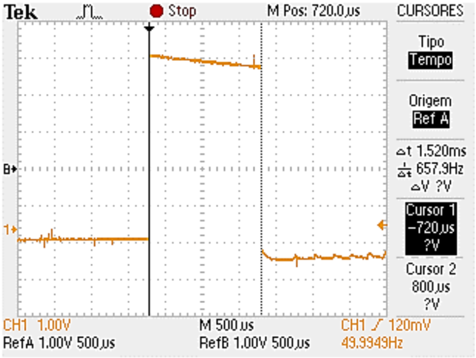

Fig. 2 Practical servomotor signal test

Regarding the perception of obstacles, we opted for the classic HC-SR04 ultrasonic sensors, widely used in robotic projects and whose technical information extracted from the manufacturer's manual can be found in Fig. 3 [5].

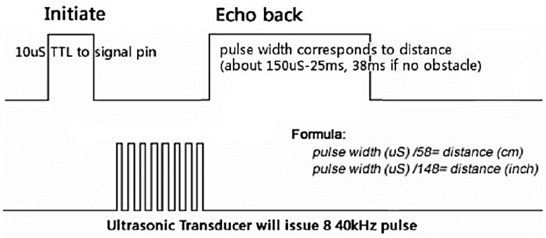

Fig. 3 Waveforms and ultrasonic sensor equations

First, it has opted for a distance of 40cm from a supposed obstacle to the mobile robot, adopted as a standard for the frontal sensors' tests. Equation (1), extracted from the manufacturer's manual of the ultrasonic sensor, shows the calculation performed for said distance. This distance is measured according to the return time of the echo signal by the sensor.

$$\text{distance (cm)} = \frac{\text{pulse width}(\mu s)}{58} \Rightarrow \frac{2320(\mu s)}{58} \Rightarrow \text{distance} = 40\text{cm} \tag{1}$$

An object was positioned at 40cm from the autonomous robot's front ultrasonic sensor in the practical test.

Fig. 4 shows the blue trigger signal sent by the microcontroller and an orange echo signal of 2.320ms proportional to the measured distance emitted by the sensor to the microcontroller. There is excellent accuracy in the result obtained, which confirms the quality of the chosen device.

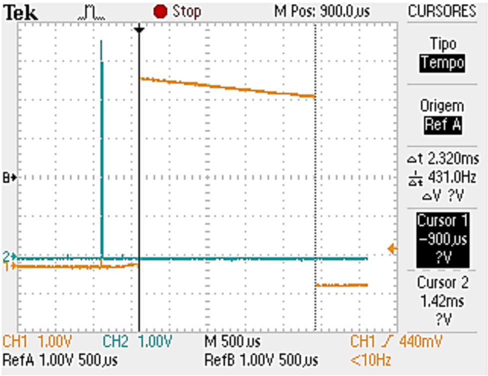

Fig. 4 Ultrasonic sensor signal tests

3. Paraconsistent Annotated Evidential Logic Eτ

The precursors of this type of logic are the Polish logicist S. Ja'skowski and the Brazilian logicist N.C.A da Costa, who proposed the contradiction in the logical structure in the non-trivial logical structure, became known as the founders of Paraconsistent Logic. In this article, we consider a particular paraconsistent logic: the Paraconsistent Annotated Evidential Logic Eτ. The language of this logic is based on propositions of type $p\ (\mu, \lambda)$, where p is a proposition and (μ, λ) indicate the degrees of favourable evidence and contrary evidence, respectively.

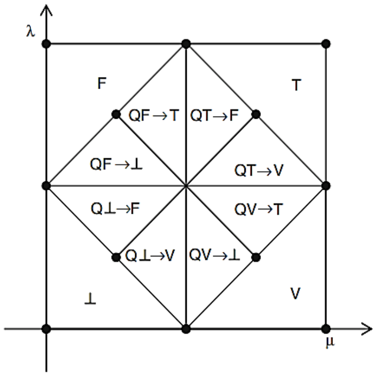

Fig. 5 Lattice of making-decision

The pair (μ, λ) is called the annotation constant, with the values of μ and λ limited between 0 and 1 [9]. Input data processing involves applying minimisation and maximising connectives to atomic formulas A and B, defining the resulting state. The formulas A and B consider the propositions with their respective degrees of certainty and uncertainty $pA\ (\mu_1, \lambda_1)$ and $pB\ (\mu_2, \lambda_2)$. Initially, the highest value is obtained between the degrees of certainty $(\mu_1\ OR\ \mu_2)$ obtaining the resulting degree of certainty (μ_R) then minimising the degrees of uncertainty $(\lambda_1\ OR\ \lambda_2)$ obtaining the resulting degree of uncertainty *(λ_R)*. The resulting logical states vary towards the four extreme logical states: True, False, Inconsistent, and Paracomplete. The concept underlying this process can be understood using a bi-dimensional graph representing the final propositions, as can be seen in Fig. 5. The vertices in the lattice represent the four extreme logical states.

The logical output states represented by internal regions in the lattice that is not the extreme logical states are called non-extreme logical states. Each non-extreme logical state is named according to its proximity to the extreme logical states, and symbology and its meaning can be observed in Table 1 [10].

Table 1 – Symbology and Nomenclature of Non-Extreme States

Non-Extreme States	**Symbol**
Quasi true tending to Inconsistent	QV→⊤
Quasi true tending to Paracomplete	QV→⊥
Quasi false tending to Inconsistent	QF→⊤
Quasi false tending to Paracomplete	QF→⊥
Quasi Inconsistent tending to True	QT→V
Quasi Inconsistent tending to False	QT→F
Quasi Paracomplete tending to True	Q⊥→V
Quasi Paracomplete tending to False	Q⊥→F

The Paraconsistent Annotated Evidential Logic Eτ has been standing out over the last few years as a powerful tool that can be used in dynamic system control, software development, and robotics, which focuses on this work. The authors [11] developed a prototype tricycle robot based on previous experiments that used two ultrasonic sensors to detect obstacles and programming based on Annotated Paraconsistent Logic. According to the authors, the modifications made both in the structural and electronic parts of the autonomous mobile robot allowed a significant improvement in the performance of the robot's movement with a reduction in the number of collisions during the tests. Another application using Paraconsistent Annotated Evidential Logic Eτ in automated cars has stood out in the cybersecurity analysis in communication between autonomous vehicles [12].

4. Description of the Prototype

This project was conceived based on the history of the application of Paraconsistent Logic in predecessor robots and the development of robotics in autonomous navigation systems. The prototype of the project implemented with the ATmega 2560 Microcontroller can be observed in Fig. 6. The HC-SR04 ultrasonic sensors were installed correctly [13]. At the front of the robot, one observes traction motors controlled by Pulse Width Modulation (PWM). On the back can be seen the differential of this prototype compared to the predecessors, which consists of a servomotor to control the robot's direction.

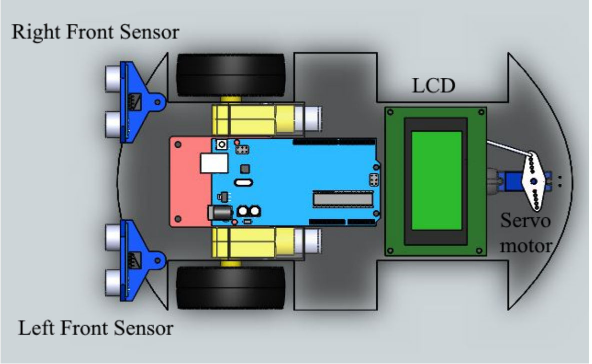

Fig. 6 Prototype of the terrestrial mobile robot

Another difference from the previous ones was the idea of using an LCD to monitor the readings of ultrasonic sensors and observe the value of the angle pointed out by the servomotor. All these observations were critical in the robot's movement tests [14].

5. Development of the Paraconsistent Annotated Logic Algorithm

Based on the fundamental concepts of Paraconsistent Annotated Evidential Logic Eτ and with the mechatronic prototype finalised, it was ideal for simulating five possibilities of positioning supposed static obstacles to the front of the robot. The criterion of this simulation was based on these obstacles in different positions to the left and right sides of the robot. Next, a normalisation of frontal sensors' readings for the values of μ and λ of the lattice was made, as can be observed in Equation (2) and Equation (3).

The normalisation process consists of adapting the distances' values obtained from the sensors and converting them to a range from 0 to 1, conceptual to paraconsistent logic [10].

$$\mu = \frac{\text{Sensor Left}}{200} \tag{2}$$

$$\lambda = 1 - \left(\frac{\text{Sensor Right}}{200} \right) \tag{3}$$

Using the proposition p "The robot's front is free," paraconsistent logic concepts were applied. For each situation, it was calculated certainty's degree, according to Equation (4), and uncertainty's degree, according to Equation (5).

$$\text{Degree of Certainty} = (\mu - \lambda) \tag{4}$$

$$\text{Degree of Uncertainty} = (\mu + \lambda - 1) \tag{5}$$

It was noticed that the degree of uncertainty generated very peculiar values to be used directly in the set point of the servomotor. Then, with other values, six new tests were performed to define the robot's behaviour concerning supposed obstacles, simultaneously positioned at the same distance to the left and right frontal sensors. Table 2 shows the development of simulations and the results obtained in each case.

Table 2 - Simulation of obstacles in different positions

	Front Sensors				**Degree of uncertainty**	**Setpoint (º)**
Situation	**Left (cm)**	**μ**	**Right (cm)**	**λ**		
1	10	0.2	50	0	-0.8	-75.76
2	20	0.4	40	0.2	-0.4	-37.88
3	30	0.6	30	0.4	0	0
4	40	0.8	20	0.6	0.4	37.88
5	50	1	10	0.8	0.8	75.76

Table 3 shows a gradual change in the degree of certainty for these new cases that ranged from 0.99 to -0.70. These values were used as decision-making for speed control and braking. The control values obtained in the simulations were applied to the paraconsistent algorithm programming developed in the C Language directly in the Interface Development Environment (IDE) of Arduino ATmega 2560.

Table 3 - Simulation of obstacles in equal positions simultaneously

	Front Sensors				**Degree of certainty**
Situation	**Left (cm)**	**μ**	**Right (cm)**	**λ**	
1	180	1	180	0.01	0.99
2	150	0.75	150	0.25	0.50
3	120	0.60	120	0.40	0.20
4	90	0.45	90	0.55	-0.10
5	60	0.30	60	0.70	-0.40
6	30	0.15	30	0.85	-0.70

The program's algorithm was divided into four main blocks to facilitate its implementation, the block of the frontal sensors, the block of paraconsistent logic, the control block of the servomotor, and the control block of speed and traction.

```
// Front Sensor Block
trigpulse_1(); //calls the function trigger of the right front sensor
pulse_1 = pulsein (echo_1, high);
rt_ft_sr =pulse_1/58; //calculates obstacle distance to right front sensor
trigpulse_2();//calls the function trigger of the left front sensor
pulse_2 = pulsein (echo_2, high);
lt_ft_sr =pulse_2/58; //calculates obstacle distance to left front sensor
if(rt_ft_sr >=50) { rt_ft_sr =50; } //limits distance measured at 200cm
if(lt_ft_sr >=50) { lt_ft_sr =50; } //limits distance measured at 200cm

//Paraconsistent Logic Block
mi = (sr_fe/50); //process of normalization of favorable evidence µ
la = (1-(sr_fd*0.02)); //normalization process of the contrary evidence λ
deg_unc = ((mi+la)-1); //calculates the degree of uncertainty
deg_cer = (mi-la); //calculates the degree of certainty

//Servomotor Control Block
sv_set_pt = 538.42*gra_inc+551.5; //calculates the set point of the servomotor
ser_pos = map (sv_set_pt, 0 , 1023, 0, 180); //positions the servomotor

//Speed and Traction Control Block
pwm_set_mt = deg_cer*105 + 150; //calculates the pwm of the traction motor
analogwrite  (rt_trc_mt, pmw_set_mt); //controls right motor traction
analogwrite  (lt_trc_mt, pwm_set_mt); //controls left motor traction
 if (deg_cer > -0.9)     {
digitalwrite (in1_mot_dir, high); //traction motors follow forward
digitalwrite (in2_mot_dir, low);
digitalwrite (in3_mot_esq, high);
digitalwrite (in4_mot_esq, low); }
else if(deg_cer <= -0.9) {
digitalwrite (in1_mot_dir, high); //brake traction motors
digitalwrite (in2_mot_dir, high);
```

```
digitalwrite (in3_mot_esq, high);
digitalwrite (in4_mot_esq, high);}
```

6. Experimental Results and Discussion

Experimental tests in a natural environment are designed to find the optimal speed of robot displacement and maintain the ability to avoid collisions with obstacles. During the tests, it was found that the robot's prototype can develop a maximum speed of 1.25m/s. The first circuit served to test the robot's ability to move through a narrow passage, positioned to the robot's left. This test showed many collisions with the obstacles because of the very high speed. Thus, other tests were performed on the same circuit at reduced speed. Fig. 7 records the robot's trajectory with a red line, in which, at point A, the right frontal sensor detects the presence of an obstacle, diverting the robot to the left. At point B of the trajectory, the left frontal sensor detects the other obstacle's presence and corrects the robot's trajectory to the middle of the narrow passage. At point C, it can be seen the robot moving with the corrected trajectory.

Therefore, there was an excellent ability to avoid collisions at a speed of 0.5m/s. The second circuit served to test the robot's ability to move through obstacles that maintained a symmetrical distance from a central axis. A standard distance of 40cm from one obstacle to the other was adopted, and thus it was observed the curve of the mobile robot's trajectory concerning this axis.

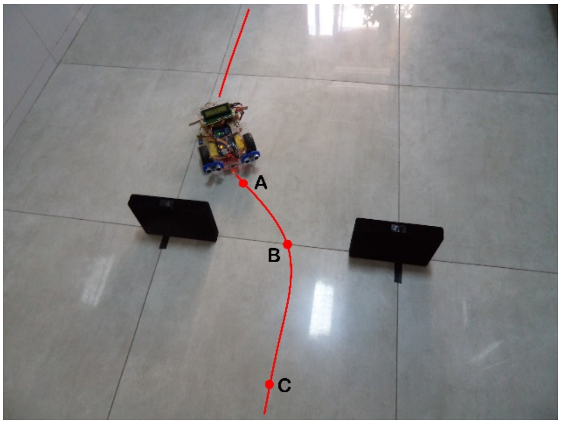

Fig. 7 Obstacle and a narrow passage to the right of the robot

In Fig. 8, the red line notes the robot's trajectory, in which, at point A, the left frontal sensor detects the presence of an obstacle, diverting the robot to the right. At point B of the trajectory, the right frontal sensor detects the other obstacle's presence and corrects the robot's trajectory to the middle of the passage. At point C, it can be seen the robot moving with the corrected trajectory. For this new test condition, an excellent performance of the robot displacement at a maximum speed of 0.6m/s was verified without collisions.

The results obtained from robotic navigation through the paraconsistent algorithm point to the possibility of acting in conjunction with other navigation techniques. In work developed by the authors [5], which consists of a system to avoid collisions with obstacles, the information of the prototype's frontal ultrasonic sensors could also be used by the paraconsistent algorithm proposed in this work. So get one more way to avoid collision with obstacles. Similarly, the work developed by the authors [4] developed a robotic navigation system by recognition of artificial landmarks, and the paraconsistent algorithm could act as an extra resource to avoid collisions with obstacles. The paraconsistent algorithm can also be implemented in the prototype proposed by the authors [6]. They built a robot for interior odometric navigation combined with previously realised images. The authors verified that odometric navigation has inaccuracies, which can be minimised with navigation aid with paraconsistency proposed in this work.

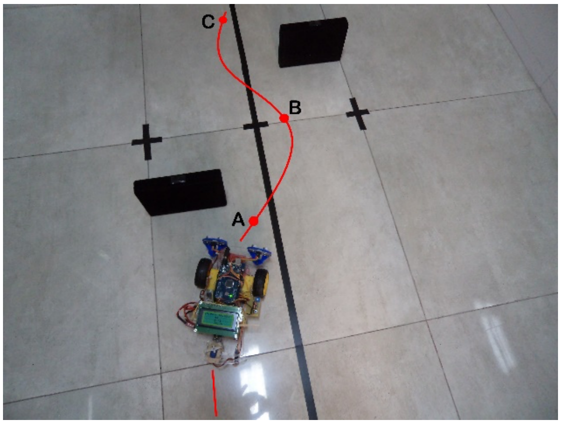

Fig. 8 Left obstacle and symmetrical passage

The paraconsistent algorithm can also assist decision-making during navigation with the same intention of avoiding collisions. The tests also demonstrate improvements in displacement performance and precision of curve movements compared to predecessors developed by the authors [12]. This is because of the servomotor acting in conjunction with the paraconsistent algorithm developed in the C Language.

7. Conclusion

The work showed that applying the Paraconsistent Annotated Evidence Logic Eτ in algorithms was practical and can be done directly without the need for extra-logical devices, even in the presence of conflicts or lack of information. In this way, it expands the contribution to several robotic navigation systems for decision-making to avoid collisions with obstacles. The paraconsistent algorithm developed in the C Language proved to be efficient and easy to integrate with other systems. Besides, using a servo motor to control the robot's direction ensures agility, speed, and precision in movements. We hope to contribute more to the use in forthcoming works. As future works, Paraconsistent Logic can be extended to other uses in navigation systems, as positioning in the map.

Acknowledgements

"This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001"

References

[1] Mohamed, S. A., Haghbayan, M. H., Westerlund, T., Heikkonen, J., Tenhunen, H., and Plosila, J. (2019). A survey on odometry for autonomous navigation systems. IEEE Access, 7, 97466-97486.

[2] Cheng, Y., Maimone, M., and Matthies, L. (2005). Visual odometry on the Mars exploration rovers. In 2005 IEEE International Conference on Systems, Man and Cybernetics (Vol. 1, pp. 903-910). IEEE.

[3] Du, S., Sun, W., and Gao, Y. (2014). An investigation on MEMS IMU error mitigation using rotation modulation technique. In Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014) (pp. 1822-1838).

[4] Zhong, X., Zhou, Y., and Liu, H. (2017). "Design and recognition of artificial landmarks for reliable indoor self-localisation of mobile robots". International Journal of Advanced Robotic Systems, 14(1), 1729881417693489.

[5] De Simone, M. C., Rivera, Z. B., & Guida, D. (2018). Obstacle avoidance system for unmanned ground vehicles by using ultrasonic sensors. Machines, 6(2), 18.

[6] Atali, G., Garip, Z., Karayel, D., and Ozkan, S. (2018). Localisation of mobile robot using odometry, camera images and extended Kalman Filter. Acta Physica Polonica A, 134(1), 204-207.

[7] Gheorghe, G. (2018). The design and control of a mobile vehicle with ARDUINO microcontroller based on algorithm PID. Annals of Constantin Brancusi University of Targu-Jiu. Engineering Series, (1).

[8] David, A. J., Arturo, M. E., and Fernando, P. P. (2017). Low-Cost Servomotor Driver for PFM Control J. Sensors, 18(1).

[9] Abe, J. M., Akama, S., and Nakamatsu, K. (2015). Introduction to annotated logics: foundations for paracomplete and paraconsistent reasoning (Vol. 88). Springer.

[10] Abe, J. M. (Ed.). (2015). Paraconsistent intelligent-based systems: New trends in the applications of paraconsistency (Vol. 94). Springer.

[11] Torres, C. R., Abe, J. M., Lambert-Torres, G., da Silva Filho, J. I., and Martins, H. G. (2009). Autonomous mobile robot Emmy iii. In New Advances in Intelligent Decision Technologies (pp. 317-327). Springer, Berlin, Heidelberg.

[12] Fernandes, M. B., Barbin, S. E., Abe, J. M., & Lima, L. A. (2019). “Analysis Of Cyber Security In Autonomous Vehicles Using Paraconsistent Logic”. Iberoamerican Journal of Project Management, 10 (1), 30-48.

[13] Tedeschi, A., Calcaterra, S., and Benedetto, F. (2017). “Ultrasonic radar system (URAS): Arduino and virtual reality for a light-free mapping of indoor environments”. IEEE Sensors Journal, 17(14), 4595-4604.

[14] Hexmoor, H. (2013). Essential Principles for Autonomous Robotics (Synthesis Lectures on Artificial Intelligence and Machine Learning). San Rafael, CA, USA: Morgan & Claypool.