Nat. Volatiles & Essent. Oils, 2021; 8(5): 1665 - 1674

Neural network-based optimization of intelligent odometry system in an autonomous robot for accurate position and orientation estimation

[1]Dr. Anandakumar Haldorai, [2]Dr.K.Balamurugan, [3]Dr. M. Arulaalan, [4]Mr. GD Vignesh, [5]Dr.Ram Subbiah, [6]Dr. S. Lakshmi narayanan,

1 Professor (Associate), Department of Computer science and engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India, 641202. anandakumar.psgtech@gmail.com
2 Assistant Professor, Department of Electrical and Electronics Engineering, Sri Ramakrishna Engineering College Coimbatore. kb.sreceee@gmail.com

3 Professor & Head, Department of ECE, C K College of Engineering and Technology, Cuddalore. ecehod@ckcet.com

4 Assistant professor, St Joseph's College of engineering, Anna University, Chennai. gdvignesh1992@gmail.com

5 Associate Professor, Mechanical engineering, Gokaraju Rangaraju institute of Engineering and Technology, Hyderabad. ram4msrm@gmail.com

6 Associate Professor, Department of ECE, Gojan school of Buisness and Technology, Chennai. s.lakshminarayanan@gojaneducation.com

ABSTRACT

The purpose of this study is to determine the accurate orientation and position of the intelligent dummy system for the Verdino autonomous robot. The position and orientation of the autonomous robot are calculated using an odometric mathematics model derived from the robot's trajectory equations. The odometry encoders are the system inputs, and the wheel diameter and distance are used to parameterize the model. This model is used to get the optimum nominal parameters by doing a minimal square minimization. This model is updated with a measurement of the real-time wheel diameter to improve the accuracy of the results. Based on data, an odometric model is trained using a neural network model. This neural network is put to the test, and the results indicate that by employing a smart odometry system, the neural network can improve estimate accuracy.

Keywords: Autonomous robot, Neural Network, Odometry system, Position, and orientation estimation.

1. Introduction

Self-localization, i.e. the autoallocation and orientation of a vehicle/robot over time is one of the major problems lately highlighted in autonomous system applications. A platform must continually retain and posture information for autonomous navigation, obstacle avoidance and object tracking. The odometry estimate involves the computation of relatively different poses between two or more sensor data frames, as regards translations and rotations. It monitors the autonomy and is followed by a procedure that integrates these changes in terms of position and with respect to a beginning state, resulting in a global position

Wheel odometry is one of the key types of self-containment of robots utilised as 2 and 4-wheel robots for various skid-steel robots. The right and left wheels may be moved in various directions and velocities, regardless of vehicles. A well-known example of this kind of robotics is NASA's Mars Exploration Rover (MER)[1]. They are suitable for a range of applications. It's important. The wheel odometry technique utilises robot-mounted encoders to track every wheel's number of rotations. The number of rotations included in a dynamic model describing the location of the existing robot in respect to the beginning point [2].

Different location techniques including GPS, laser, markers for radio frequency, natural or artificial beaches and others[3] utilise odometry in many ways. [4]. If mechanical odometry is not possible, optical odometry[4] may be used to determine the location of the vehicle based on picture differences. A complex sensor system is built on many sensors to achieve position and orientation[5] for an independently polluted vehicle. In certain methods, Kalman filters [7], Monte Carlo-based particle filters [8,] and others are also utilised. Sensors without odometry often need external environmental information, thus in many circumstances they are useless. For example, when a vehicle views the sky, the GPS service is lost[9]. When we place beacons, it is necessary to structure the whole environment, otherwise a laser may detect the beacons in the environment, and these features must, like prior occasions[10], be in the field of laser activity. The work's major addition is the use of an autonomous robot sensor set to enhance the accuracy by monitoring the roll diameter in real time. In order to anticipate the location and direction of the autonomous robots, input from the sensor is treated through neural network-based optimisation.

2. Related works

In order to find things via sound, a sound based locale is utilised. But for non-sound generating items, this method is not relevant. [11] a multi-sensor location system (MSS) that is a deterministic method of interval analysis for mobile robots has been created. The robots are used as sensor technology using ultrasonic sensors. [12] utilised GPS-based system suitable to the location of urban robots. For mobile robot location estimation, a probabilistic grid method is suggested. The proposed approach is based on Bayesian design and is suitable in robots, albeit noisy ultrasonic sensors and occupancy grid maps are present, which are approximate representations for the environment. The Markov model (POMM) is used to locate the robot partly in interior environments in particular in offices. In line with the topological environment structure the state space is structured according to this approach. The grid size must, however, be specified for this model.

[13] Complementary filtering (CF) for the location and orientation estimate of mobile robot by combining GPS, digital compass and inertial unit measurement data (IMU). This leads to cheap implementing costs and a satisfactory outcome for outdoor robot navigation. [14] introduced an intelligent space-based TCP/IP client server system that locates and monitors various robot kinds, including wheelchairs for robotics. The system suggested has 11 synchronous photography cameras.

Each of these cameras and robot is connected to the server. The server gathers, analyses and transmits data from different sensors to the customer. A method of calibration using the multi-camera network is proposed for localisation. First, the robot flows across the whole camera field of view on a calibration model. Next, robotic odometry and pictures collected are used in the technique of calibration suggested. This provides a method for locating and calibrating several cameras concurrently. [15] presented a hybrid method to location that incorporates adaptive MCL (AMCL), iterative closest point (ICP) and discrete transforming Fourier (DFT). In realtime settings such as industry, our hybrid method offers a sub-centimeter precision in place. In order to provide the estimate of the robot position, the AMCL algorithm combines laser range measuring with data on odometry. The following is an early approximation of the ICP algorithm. The ICP algorithm's output is finally fed into a DFT method that gives the end result. [16] developed an enhanced localization method based on the concept of multilateration. [17] suggested a method of effective localisation to resolve posture monitoring and kidnap recovery concurrently.

3. Smart system of odometry

The system of odometry is based on robot wheel combined encoders. Every encoder gives 1024 pulses per revolution and every wheel rotation produces one encoder revolution (1:1 coupling). Rolling rotation is communicated via a flexible mechanical transmission system from the centre of each wheel to the encoder on the side of the solo robot. The encoder output is attached to ad hoc electronics, which every 0.5 ms samples the encoder signal. The electronics are built for the measurement and integration of the encoder signals and all integration periods of 20 ms are sent to the on-board computer. Based on the Euler integration, which gathers encoder increments throughout the inclusion time, the integration is carried out in the ad-hoc electronic microcontroller.

The orientation of a robot's cinematic state (position) in respect to a fixed reference system is represented by position (x, y), in relation to (angle formed between the reference system's X axis and the robot's longitudinal location). A circular trajectory is followed while the prototype is rotating. The period of integration is sufficiently short to keep trajectory curvature constant. Figure 1 shows the first, (x_i, y_i) and the last, (x_f, y_f) post-integration time. The displacements of robot wheels are derived from the robots and the size of the wheel.

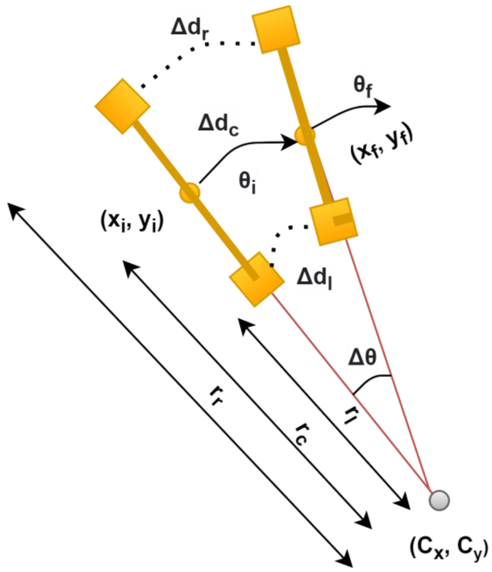

Figure 1: A smart model system odometry.

The arc length of right (Δ_{dr}) and left (Δ_{dl}) encoder measurements may be used to calculate robot wheels.(Δ_{cr}, Δ_{cl}), wheel radius (R_r, R_l), and encoder resolution (Enc_r, Enc_l).

$$\Delta_{dr} = \frac{2\pi R_r \Delta_{cr}}{Enc_r}$$

$$\Delta_{dr} = \frac{2\pi R_l \Delta_{cl}}{Enc_l} \qquad (1)$$

Radius is determined based on wheel distance for each wheel and centre.d_w.

$$r_r = \frac{\Delta d_r d_w}{\Delta d_r - \Delta d_l}$$

$$r_l = \frac{\Delta l_r d_w}{\Delta d_r - \Delta d_l}$$

$$r_c = \frac{d_w}{2} \frac{\Delta d_r + \Delta d_l}{\Delta d_r - \Delta d_l} \qquad (2)$$

Angle Δ_θ and position (Δ_x, Δ_y) is calculated based on these assumptions' increments are:

$$\Delta_\theta = \frac{\Delta d_r - \Delta d_l}{d_w}$$

$$\Delta_x = r_c(\cos(\theta)\sin(\Delta_\theta) - sin(\theta)(1 - cos(\Delta_\theta)))$$

$$\Delta_y = r_c(\sin(\theta)\sin(\Delta_\theta) + \cos(\theta)(1 - \cos(\Delta_\theta))) \qquad (3)$$

The location and guidance with the latest calculated increments are updated in the final step:

$$\theta_{i+1} = \Delta_\theta + \theta_i$$

$$x_{i+1} = \Delta_x + x_i$$

$$y_{i+1} = \Delta_y + y_i \tag{4}$$

Only 3 free parameters, (R_r, R_l) and wheel spacer, d_w the odometry model is controlled by. The accurate measurement of these parameters is the sole stage in adjusting the odometric system according to this method.

Another approach to deal with the issue of dynamic odometry consists in preventing the calculation and learning of data-based models from autonomous dynamic equations. Such models are very flexible, and other variables affecting odometry are taken into consideration that are not addressed in a mathematically static model. A learning model has many impacts, including speed, rotation direction, changes in the spin rate, slippage of the wheel, etc.

4. Optimizing the Neural Network

Neural networks are biological neuron-inspired algorithms. Networks may be used to learn from data complicated nonlinear mathematical functions. The collection of artificial neurons receiving an input and calculating a product output reflects the function to be learnt. In the training phases, weights are calculated in comparison to the required output. Network output. Changes in the function gradient are used to adjust weight across different iterations until the failure reaches a minimal or maximum number of iterations. Neural networks are a generic interpolator, thus a good generalization may be achieved with proper training. The neural network is used to enhance the accuracy of odometry when the errors of a mathematically determined odometric model are rectified using a neural network. In order to assess a mobile robot mistake, the neural networks are employed. This study trains neural feed networks using odometry data and uses GPS information as a fundamental truth. Figure 2 displays the network. It requires two input: the incremental number for the left and right encoders for the current and prior iteration. The activation function is sigmoid, for GPS-based encoders the training set includes 7100 data points, whereas the validation set contains 3000 points of the same image but is not utilised to train the network. Tests are pertaining to 11,500-point data in the same setting. The GPS data is interplayed to obtain the autonomous robot's current GP SPA location, such that encoder data are based on the same clock on both odometry and GPS data. The first training is carried out on-line, so that all the data are accessible for interpolation.

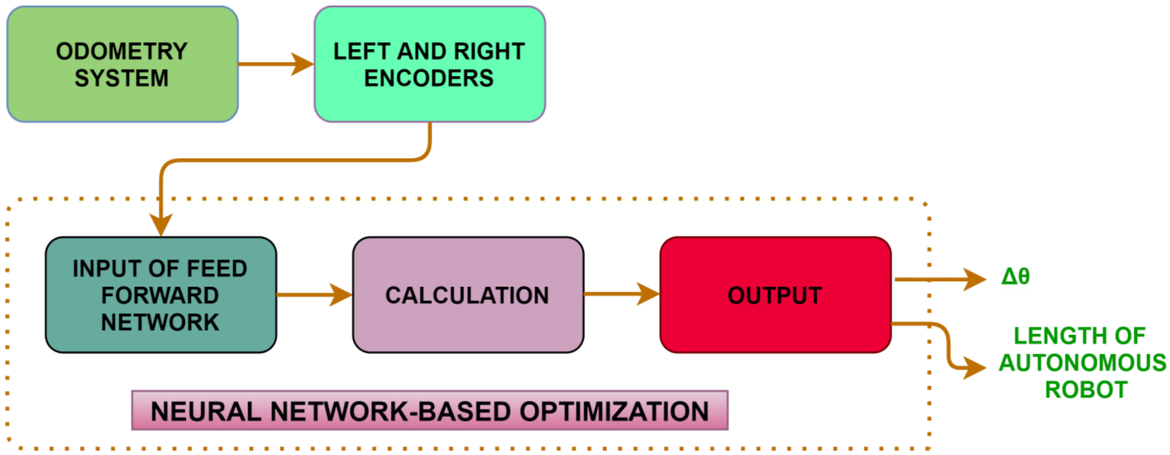

Figure 2: A neural network that feeds information back to itself.

The angle of the network's outputs changes (Δ_θ), change in length, and change of rpm sensors causes the autonomous robot to travel. The trained neural network thus acts like a modelling function, which receives input rating and provides output shift and angle.

The network is trained using a medium squared error method controlled by Levenberg-Marquadt. There are two neurons, fifty hidden neurons, and two outputs. Each odometrical input increases and shows the angle, and the training dataset is gathered from the worldwide data collection. These facts are utilised without more information to train the network.

The equation (5) is used to compute posture with the use of neural network output (X, Y, θ). The inkremental angle and shift are provided by the neural model. All incremental network outputs are combined to determine the final posture.

$$\theta_{i+1} = \Delta_\theta + \theta_i$$

$$X_{i+1} = X_i + length \cdot cos(\theta_{i+1})$$

$$Y_{i+1} = Y_i + length \cdot sin(\theta_{i+1}) \quad (5)$$

The previous neural networks are updated with some previous knowledges about the autonomous robot trajectory to enhance the accuracy of the posture computation. The new network uses four inputs: the odometry of the left and right wheels rises compared to current measures and prior measurements. This data allows the network to build up the odometry function using other data, such as speed and speed. The accuracy of a neural network model would increase with the provision of additional data. A neural network has been assessed using six inputs in the final test. Coders from the left and right rollers as well as the diameter of the wheel and sensor wheel from current or prior reading are included. For network training, GPS ground data output is utilised. The network includes all the data needed to optimise the model. As with previous studies, the number of hidden neurons and outputs is the same.

These experiments demonstrate that neural network is a strong substitute for mathematical models, improves the accuracy and adjusts actual system behaviour when sufficient information is available. The neural network may take use of the relationship between inputs and outputs, not included in the mathematical model. The model parameters vary throughout time as one of the difficulties of an actual odometric system. These modifications may be categorised as rapid or slow. A variation in weight of passengers may lead to rapid changes, thus the model changes quickly. Slow changes are caused by environmental changes, such as pressure loss of the wheel, temperature fluctuations or tear of the wheel. A static mathematical model and a learned model cannot confront this problem, and therefore the precision of the model will decrease with time. To address this problem, an ongoing training scheme is employed to update the model in real time. The location subsystem is utilised for tuning the neural odometric system while the cart is travelling.

In a historical buffer the navigation data (encoders and wheel radius) are incorporated to train the network with the last data to keep the actual system model correctly. The primary issue is obtaining adequate tracking data for network training. The prototype location subsystem is based on the fusion of several sensors and the location in a prototype map. The accuracy of the autonomous location relies on many variables and based on the information of the sensor localization quality may vary. Localization of Monte Carlo (MCL) is useful for estimating a mobile robot's position by utilising a map that shows the real posture, allowing heterogeneous sensor data to be fused. The adaptive Localization System MCL algorithm combines the input of wheel odometry, an inert unit, a global positioning system, and laser scanning. A particle weighting model using GPS data is used to improve performance in comparison to a particle making method. The results of the technique are the estimated positions and a covariance matrix representing location uncertainty.

Only feasible changes to the model are retrained by the neural network when the accuracy in the ground truth is increased (covariance less than 0.5). You may construct a smooth and advanced model modification using this technique and can get an accurate, reliable and adaptable model, but your data are not correct. All new measurements of covariants smaller than 0.5 are only applied once to the neural network, leading to extremely low cost of retraining.

5. Result and discussion

Some experiments are performed utilizing our Verdino platform to verify the equations provided. As the ground truth a differential GPS is utilised and odometry-synchronized data are acquired via GPS placement. This information is generated based on number of satellite availability and position optimization outcomes. It provides real-time information regarding position correctness. The data are collected with clear sky in excellent weather circumstances to obtain the truth from the ground. The data is examined, and this dataset is deleted if any errors are found. The data set utilized in this research had an inaccuracy of less than 1.5 cm throughout the journey.

This prediction has been tested in many studies. Each test consisted of the system moving from a fixed starter to a setpoint. There were different proportionate gains in each test. The recorded autonomous robot movements are shown in Fig. 3. Point-to-point expertise has been merged to evaluate the efficacy of orientation control on a point-by-line basis. Tracking an extra corner of an independent robot supplied the input. In order to separate translation to the limiting point from the rotation to achieve alignment, the turning point was selected as the corner utilised to define the point-to-point motion. The system positioning precision was shown to remain unaltered experimentally. The precision of the sides alignment has been shown to be less than 1.0. The cycle time fell to 9,5 Hz due to an increase in tracking load and numeric computations. The system response to a step input for different gain levels is shown in Fig. 4. The indicated values are the corners between the X-y levels of the globe.

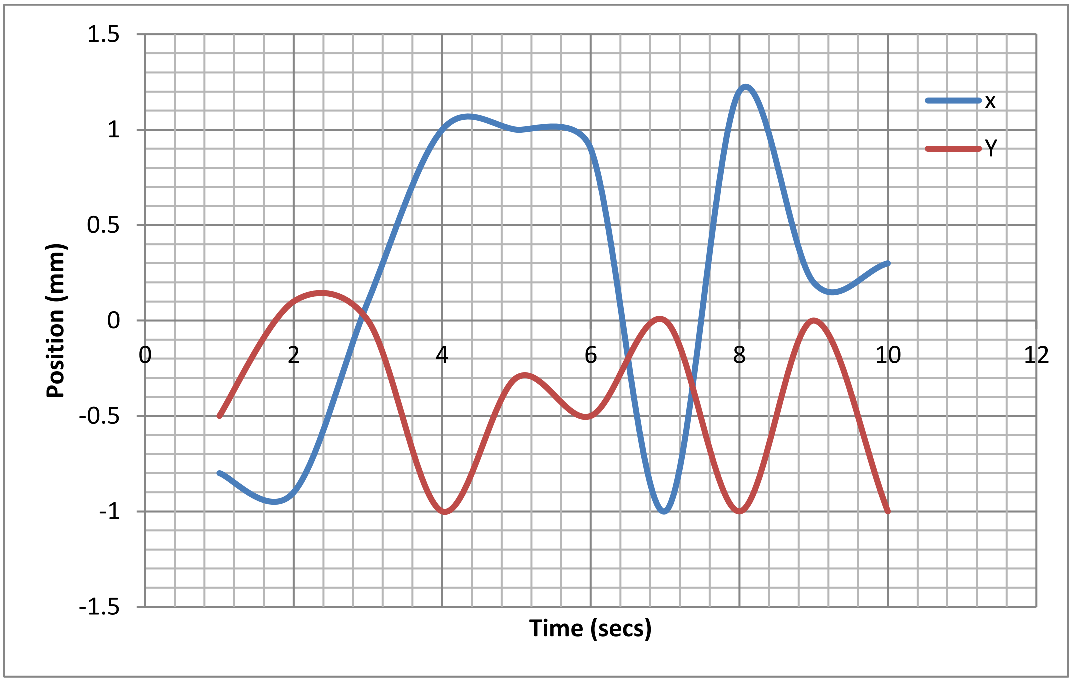

Figure 3: The point-to-point autonomous robot position.

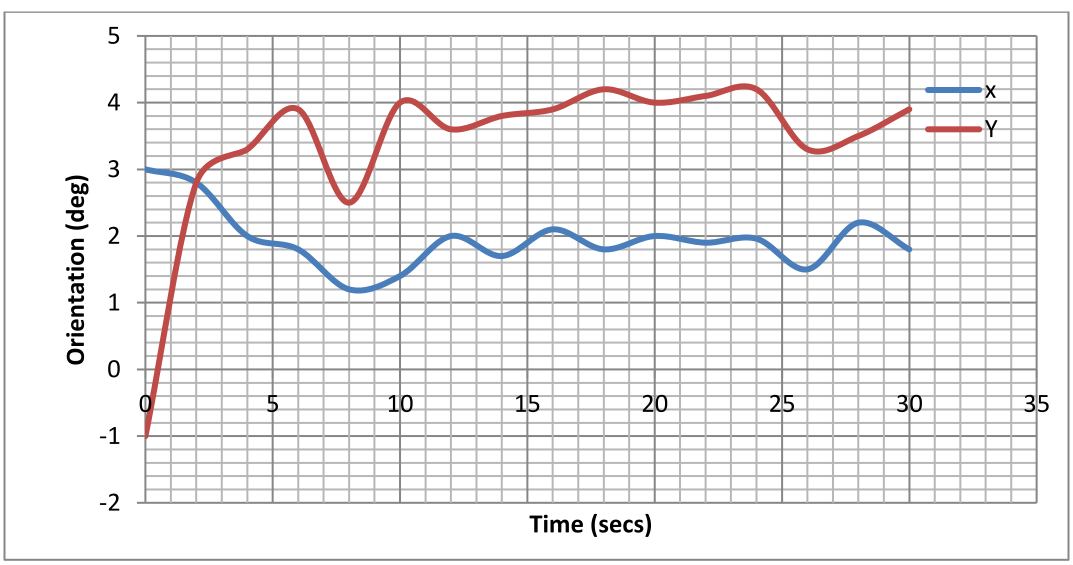

Figure 4: The orientation of autonomous robot while positioning.

6. Conclusion

One of the primary sensors for position assessment and guidance on robots and self-employed vehicles is an intelligent odometry system of an independent robot. But its accuracy tends to be poor over long distances. Positions are accurate in brief movements, but mistakes grow rapidly and as travelling distance rises, differences to the ground truth emerge. The focus of the article is on the autonomous Verdino robot odometric system and the methods used to improve its precision.

A model-based optimization of the neural network represents autonomous robot movements to enhance the odometry quality. The benefit of this model is the knowledge of data input-outcome linkages and certain effects that cannot be included in a mathematical model. Tested in three versions, this neural network template. The initial part of it is a basic output relation where just one odometry data point is taken by the neural network and angled and displaced. The precision of the location is enhanced using previous information so that factors such as speed or acceleration may be considered. The model's input is the most recent and previous odometric data for this historical data. If not only the odometry, but also the real-time range of the wheels, the best results are available when the neural network is provided. If the model is kept up to date at this point, the optimization of the neural network is constantly trained based on information on localisation. Finally, the neural network-based optimisation of the odometric system may be a very powerful system to enhance precision.

REFERENCE

1. Y. Cheng, M. Maimone, and L. Matthies, ''Visual odometry on the mars exploration rovers,'' in Proc. IEEE Int. Conf. Syst., Man Cybern., vol. 1, Oct. 2005, pp. 903–910.
2. K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, ''An overview to visual odometry and visual SLAM: Applications to mobile robotics,'' Intell. Ind. Syst., vol. 1, no. 4, pp. 289–311, Nov. 2015.
3. Espinosa, F.; Santos, C.; Marrón-Romera, M.; Pizarro, D.; Valdés, F.; Dongil, J. Odometry and laser scanner fusion based on a discrete extended Kalman filter for robotic platooning guidance. Sensors 2011, 11, 8339–8357.
4. Scaramuzza, D.; Fraundorfer, F. Visual Odometry Part I: The First 30 Years and Fundamentals. IEEE Robot. Autom. Mag. 2011, 18, 80–92.
5. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The stanford entry in the urban challenge. J. Field Robot. 2008, 25, 569–597.
6. Vivacqua, R.; Vassallo, R.; Martins, F. A Low Cost Sensors Approach for Accurate Vehicle Localization and Autonomous Driving Application. Sensors 2017, 17, 2359.
7. Bento, L.C.; Nunes, U.; Moita, F.; Surrecio, A. Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments. In Proceedings of the 2005 IEEE Intelligent Transportation Systems, Vienna, Austria, 13–16 September 2005; pp. 245–250.
8. Wang, C., Chao, K., Sivaperumal, S., Suresh, P., "Anti-PVT-Variation Low-Power Time to Digital Converter Design Using 90 nm CMOS Process" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 28, Issue 9, 2020, pp. 2069 – 2073.

9. Suresh, P., Mariyal, C., Sivasubramonia Pillai, T.V., Rajesh, K.B. and Jaroszewicz, Z., "Study on polarization effect of azimuthally polarized LG beam in high NA Lens system," Elsevier – Optik, vol. 124. Issue 21, 2013. pp. 5099-5102
10. Karthikeyan, V., Suresh, P., Saravanakumar, U., Anil Venkata Rama Sekhar, AKM. and Srikanth, S., "Wide Band Millimeter Wave Fabric Antenna for 60GHz Applications," International Journal of Innovative Technology and Exploring Engineering, Vol. 8, Issue 9, 2019, pp. 1211-1214.
11. Ashokaraj, I.A.R., Silson, P.M.G., Tsourdos, A., White, B.A., 2009. Robust sensor-based navigation for mobile robots. IEEE Trans. Instrum. Meas. 58 (3), 551–556. https://doi.org/10.1109/TIM.2008.2005266.
12. Georgiev, A., Allen, P.K., 2004. Localization methods for a mobile robot in urban environments. IEEE Trans. Rob. 20 (5), 851–864. https://doi.org/10.1109/ TRO.2004.829506.
13. Neto, A.A., Macharet, D.G., Campos, V.C.da.S., Campos, M.F.M., 2009. Adaptive complementary filtering algorithm for mobile robot localization. J. Braz. Comput. Soc. 15, 19–31. https://doi.org/10.1007/BF03194503.
14. Rampinelli, M., Covre, V., de Queiroz, F., Vassallo, R., Bastos-Filho, T., Mazo, M., 2014. An intelligent space for mobile robot localization using a multi-camera system. Sensors 14 (8), 15039–15064. https://doi.org/10.3390/s140815039.
15. Vasiljevic´, G., Miklic´, D., Draganjac, I., Kovacˇic´, Z., Lista, P., 2016. High-accuracy vehicle localization for autonomous warehousing. Rob. Comput. Integr. Manuf. 42, 1–16. https://doi.org/10.1016/j.rcim.2016.05.001.
16. Suresh, P., T Venkata Prasad, B Uday kiran Yadav, SV Narayana Reddy, S Yaswanth Reddy, R.Rajkumar, U.Saravanakumar, 2020 "Dynamic Partial Reconfiguration Performance Analysis In Field Programmable Gate Array For Streaming Applications," International Journal of Scientific & Technology Research, Vol. 9, Issue 02, 2020, pp. 3098-3102.
17. Choi, J., Choi, M., Chung, W.K., 2012a. Topological localization with kidnap recovery using sonar grid map matching in a home environment. Rob. Comput. Integr. Manuf. 28 (3), 366–374. https://doi.org/10.1016/j.rcim.2011.10.008.