Map Building of Indoor Environment with Sensors using Neural Network

[1]Dr.S Angel Latha Mary,
Department of Computer Science and Business Systems
Sri Eshwar College of Engineering
Coimbatore, India
xavierangellatha@gmail.com

[2]Dr.K.Ulagapriya
Department of Computer Science and Engineering
Vels Institute of Science Technology and advanced studies
Chennai, India
upriya.se@velsuniv.ac.in

[3]Dr. A Poonguzhali
Department of Electronics and Communication Engineering,
Sri Sairam College of Engineering
Bangalore, India
poonguzhalimadhavan02@gmail.com

[4]Dr.R.Menaha
Department of Information Technology
Sri Eshwar College of Engineering
Coimbatore, India,
rmenahasenthil@gmail.com

[5]Dr Beaulah David
Department of Information Technology
Hindusthan College of Engineering and Technology
Coimbatore, India
beaulahdavid.phd@gmail.com

[6]T.R.Priyadharshini
Department of Information Technology
Hindusthan College of Engineering and Technology
Coimbatore, India
cse.priyadharshinitr@gmail.com

2023 Winter Summit on Smart Computing and Networks (WiSSCoN) | 979-8-3503-3405-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/WISSCON56857.2023.10133850

Abstract--The necessity of a blueprint of a building structure is a mandatory requirement for any reconnaissance or rescue operations. In our project, we build a modular system combining sensors related to sonar, laser, micro-wave to read sensory values and generate a 2D path of any building. The data is fetched and stored to feed to anOptimal Neural Network (ONN)-based computing system to create a 2D route with minimal discrepancies of error. Here the NN architecture is fine-tuned using Modified Dolphin Partner Optimization (MDPO) Exploration of unknown environments and space using autonomous vehicles has recently gained good attention in the field of Robotic Mapping. The recent advancements in the Internet of Things have enabled us to establish an ideal frame of reference for sonar and lidar-based systems. New effects are displayed by the sensors according to the physical characteristics of a room. The range data from sensors in various surroundings are interpreted by NNs. The distorted errors due to the material medium, particles, and moving objects present in the environment pose a threat to building a high-quality path map. The sensor fusion technique is applied to the rotatable modular array sensor to minimalize discrepancies caused by cloth materials during sonar readings, particle noise in an environment for Lidar reading, and moving human bodies present in the environment for path building and obstacle detection.

Keywords:OGM processing and building, Neural Network

I. INTRODUCTION

Map building is critical in control architectures that use planned motions, which do not require prior environmental awareness. The robot is able to make an internal model of its work environment by exploring it. The choice of how to show an internal map depends on many things, but mostly on the environment itself. A measurement map provides a more compact way to show how a slightly elevated indoor environment looks. At any given time, squared grid maps are utilized to show this type of environment [1]. A multilateral map can be made by putting together various local perspectives of the environment, which the neural network interprets as probabilities.

Where there is no real warning of the terrain, map construction and path planning are crucial for disaster and fire emergency scenarios [1]. A metric map is the more condensed depiction of the environment for internal high-density environments, and we utilise the squared grid map to describe the environment. The NN used to determine the probability values is used to generate a local map using a spinning sensor array that takes into account cell probability [2]. By combining many local perspectives of the environment that a NN interprets as probabilities, are able to create a global map. When the buildings have complicated interior structures, creating maps and paths by hand and reconstructing them using design structure data is a laborious and time-consuming operation. Data on blueprints and paths can be obtained in a variety of ways, including time-of-flight cameras and sensor fusion. In this research, use the sonar readings from the ultrasonic sensor to create an input feature map and then use the sensor fusion method with LIDAR and accelerometer sensor information for 3-DOF position tracking using a Kalman filter extension (EKF). The estimated 3-DOF pose is utilised to speed up the map realignment process and provides a 2-D approximation of the system pose. Here MDPO based optimal neural network is structured to develop an effective map building of indoor environment with sensors.

The featured points acquired from 2D pose estimation will be integrated into a global map by combining high relative probability data. Then, we are using the readings from microwave radar devices to minimalize errors that arise due to the movement of human bodies and also to detect the same during disaster-prone situations.

II. RELATED WORK

Mapping and path planning of indoor structures using robotics involves generating sensory readings and building occupancy grid maps [3]. This research develops a technique for mapping the entire vehicle organizational climate using ultrasonic sensors. Acoustic sensors display various behaviours based on the actual characteristics of the surfaces that make up the environment where the bot is traversing [4]. The range measurements of motion sensor in the various settings are decoded by a NN. This network that provides the cell occupancy probability creates a local map with squared cells [5]. Finally, the Bayes' rule is used to create a global map that incorporates many environmental viewpoints [6]. In this work, the outcomes of the method's application to construction projects in surroundings with specular surfaces and rough walls are shown. This paper's key contribution is the creation of a reliable system for creating maps in various natural settings without needing to retrain the neural net every time the environment changes. Two extreme examples, one with clean walls and the other with rough ones, where the sensors behaved almost perfectly and provided a range that was close to the actual distance, were examined to confirm the approach. A Hasselblad Range Systems with transducers from the 7000 and 600 series as well as the robots RWIB21 and Robuter II were used to get experimental data [7]. The mobile robot's use of sensors to examine the environment and create an environment model is the end objective of robot investigation. Mobile robots are anticipated to accomplish it online in the interim. Numerous approaches have been used to solve this modelling issue, including the occupation uniform grid (OGM), geometrical, topological, and 3D data [8]. Because of its excellent fast computation and favourable effects on the environment, the OGM has emerged as the most established and popular simulation environment. The techniques can be used with a wide range of sensors, including depth cameras, sonar sensors, and laser sensors. The grid approach still has several issues, despite its broad success. Every grid is thought to be independent. The probability that a grid is occupied depends only on whether a sensor has noticed it [9]; it is unrelated to the observations of nearby grids. In other words, despite the fact that adjacent grid is populated, it remains unidentified if the pattern is not observed. Because obstacles in real-7 physical environment have distinct shapes, the grids really aren't independent of one another. This presumption frequently causes the indeterminate area on the map to grow, especially when the rangefinder beam is sparse or fails most of the time. This causes a quantification gap in the accommodation map, which greatly reduces the precision of the map [10]. The typical solution is to increase the monitoring width from every beam by adding intervention to the angle of the rangefinder's beams. The inhabited area will overlap the uninhabited area because this technique diminishes the area of unfamiliar but it is not genuine.

In Existing systems, Ultrasonic sensor used for generating data to accumulate measurement data of environment objects in local map [11]. Local map construction using sensor measurement with NN gives occupancy probability for each cell. Global Map generated has less accuracy with position error due to robot's odometry as they are not fortified with the model. Raw Feed data from sensor result to arise in noisy error probability due to no filtering. Ultrasonic Sonar range readings are vulnerable to many factors such as wide angle of their radiation lobes, multiple reflections (specular reflections), diffuse reflections, and fluctuations in the propagation medium as it will arise in range reading error caused by sonar wave [12].

This paper proposes, Sensor Fusion with ultrasonic, Lidar to get more approximation on measurement data and occupancy states. Applying RBPF and Kalman Filter to reduce noisy error and provide optimal estimation of the raw sensor data. Using NN[17] to predict the occupancy probability of each cell to construct a 2D map.

III. PROPOSED METHODOLOGY

Setup of the sensory array module to fetch measurement readings and process it into dataset and using optimal estimation algorithm with the data and using the NN to predict the occupancy probability of each cell to construct a 2D map.

Sensor Device Setup

Sensor Setup involves mounting of sonar and lidar range finder to an array module made to be rotatable in both horizontal and vertical positions. Inputs pins from the sensors are connected to different digital pins of the arduino nano. SDA and SCL pings from the motion tracking mpu 6050 and VL53L0X sensor is connected to the analog pins A4 and A5 of arduino nano [13]. Sensor Array module operated to rotate 180 deg in x axis and fixed at a specific angle in y axis to generate the object position and range in the environment with the help of motion tracking sensor and range finders. Microwave sensor operated to detect the human movement in the environment. Raw data read from the sensor array is feed into the serial output from arduino controller to pc.

Data Processing

Serial Data from arduino controller will be read to python script and initially it checks for available com ports in the pc and reads the data. Serial Data is stored into a csv file in the format of sensor readings of 10 columns. Each data from different sensor is passed to filtering algorithms to get a noiseless sensor data [14].

Data Filtering

Data Filtering involves optimally estimating the raw sensor with the use of Rao-Blackwellized Particle Filter (RBPF) and extended Kalman Filter(EKF) [15]. With

the input reading at a generated time the State Estimations are calculated with each iteration of sensor readings and optimal future state values are generated. Optimally Estimated data shows the error calibration from the ground truth of the environment and the sensory reading.

EKF: To derive values for correlations, the EKF linearizes the condition in around present mean and covariance by applying partial derivatives of the procedure and observation variables. When dealing with non-linear connections, estimates can be computed by linearizing the prediction around the present value using the partial derivatives of the system and assessment variables, which is similar to a Taylor series. Let us accept that procedure again has a state vector $z \in \mathcal{R}^n$, but that the procedure is now governed by the non-linear stochastic difference equation,

$$\mathrm{x_k} = \mathrm{nf}(\mathrm{x_{k-1}}, \mathrm{u_k}, \mathrm{w_{k-1}})$$

with a measurement $\mathrm{z} \in \mathcal{R}^{\mathrm{m}}$that is,

$$\mathrm{z_k} = \mathit{n}(\mathrm{x_k}, \mathrm{v_k})$$

where $\mathrm{w_{k-1}}$ and$\mathrm{v_k}$once more stand in for processing and parameter variations.

In this instance, the entity at the present time step k is related to the province at the previous stage k-1 by the non-linear component nfin the partial differential equation as in x_k. Any commuting feature u_k and the negligible procedure noise w_k are included as parameters. The state and the $z_k$are related by the nonlinear function n in the measuring formula as in x_k. Of fact, in practise, one doesn't know the precise noise ratios at every particular time step. However, one can approximate the state and measurement vector without them as

$$\bar{\bar{x}}_k = nf(\bar{x}_{k-1}, u_k, 0) \text{and } \bar{\bar{z}}_k = n(\bar{x}_k, 0)$$

Where $\bar{\bar{x}}_k$is a posteriori information about the state at period k. There is a serious problem with the EKF in that following each nonlinear activation function, the probabilities (or intensities in the greater effort) of the several explanatory variables are also not necessarily normally distributed. Rather than being a rigorous state generator, the EKF is merely a linear approximation of Bayes' theorem.

RBPF: The mathematical design of the model is discovered in part by this technique as well. This requires precisely calculating the probability of the consistent states using a collection of Kalman filters for the continuous state distribution computation and a particle filter (PF) for the discrete state computation. Specifically, a recursive probabilistic combination of Gaussians is used to approximately model the posterior probability. The RBPF is useful for making better estimates during the learning phase, which is why it is so important. The Sampling Importance Resampling (SIR) filter is widely used as a PF method. To accomplish this, let will update a sample distribution that stands in for the posterior. The goal is to make use of the state vector's structure. It is not necessary to draw examples from the complete state field if some dependent connections between members of the state matrix can be analytically explained.

Assume that the transfer function is represented

by$x_k = \{x_{1:k}\ ;\ x_{2:k}\}$,

for which $p(x_{2:k}|X_{1:k}, Z_k)$can be assessed.

By utilising p(x (2:k)X (1:k),Z k), it is feasible to perform a PF on X_{1k} while notifying the corresponding particles of$x_{2:k}$.

In real-world applications, RBPFs are typically employed when one component of the information evolves linearly and Gaussianically as a function of the other component. In this situation, the Gaussian function $p(x_{2;k}|X_{1:k}, Z_k)$ can be updated using a set of ℕ KFs$x_{2;k}$. Please take note that N iterations of the KF are required at each phase for a method with N particles.

It has been demonstrated that the RBPF method significantly lowers the dispersion of the estimated errors. And because the PF is operating on a smaller proportion of the state's dimensions, we anticipate improved performance compared to the standard PF: There is a trade-off between the number of particles required to converge and the number of operations executed every particle.

Occupancy Mapping and Prediction

Occupancy Grid mapping (OGM) involves robust and unified approach to variety of problems in spatial representation and robot navigation. With the sensor data and employing probabilistic sensor interpretation model and random field representation schemes. Discrete random variable represents the occupied and free state of each cell to estimate the occupancy probability. A local map is constructed using a referencing system made up of 24×24 square cells with 12 cm sides, of which the robot occupies the 4×4 core cells.Each generated map sequence is feed into the connection of NN and produce a sequence of OGMs. OGM sequences are integrated to define a global semantic map.

The goal of a NN is to emulate the brain's pattern recognition capabilities by employing a series of techniques to find hidden connections within a dataset. In this context, NNs can refer to biological or synthetic networks of neurons. Since NNs are flexible, they can produce best results even if the input is modified, saving time and effort. Using NNs, a notion with origins in AI, has become increasingly common when creating occupancy mapping and forecasting systems.

Neurons are the building blocks of a NN, an architecture that attempts to simulate the brain's operation. What you

see before a neuron, broken down into its numerous parts. A typical NN is shown in fig 1.

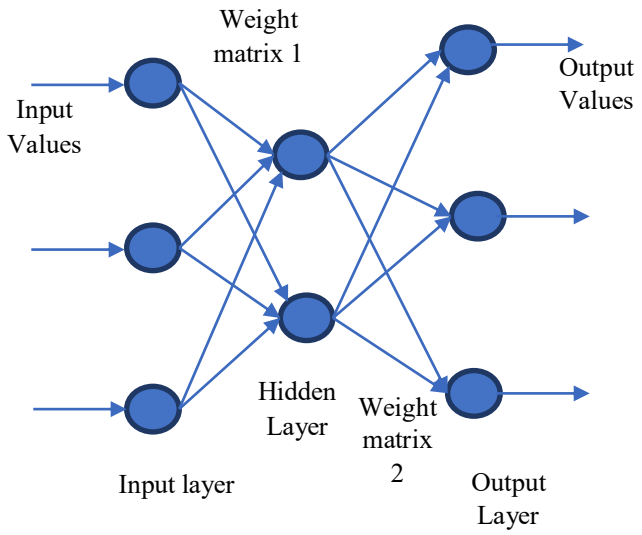

Fig.1 A typical NN

The input is the collection of features used to train the model. The input for object detection, for instance, may be a matrix of picture pixel values.Weight's primary role is to prioritise information that has a greater impact on improving learning. It accomplishes this by multiplying the input value by a scalar before applying the weight matrix.

The activation function requires a transfer function to merge many inputs into a single output value. The variables to the linear system are simply summed. The activation function makes perceptron's more flexible by allowing for inputs with varied linearity. Without it, the output would be a simple addition of the input values, preventing the introduction of quasi.Adjusting the activation function's output value is what bias is for. It acts much like a constant would in a linear function.The data that feeds the model is entered into the input layer, and it is the only layer in the full NN design that is directly visible to the user and passes the entire data from the environment without any processing.

Whatever enables machine intelligence, what it is nowadays are the hidden layers. These layers perform the bulk of the mathematics and feature extraction and sit between the input data and the final layer. Different types of data's hidden characteristics may be searched for over multiple hidden levels. Higher-level characteristics, such as edges, forms, and boundaries, are the purview of the initial hidden layers in image processing, for example. Later hidden layers, while on the other side, are responsible for more complex tasks like object recognition.

The output layer is accountable for synthesising the information learned in the previously hidden layers into a single, actionable prediction. This is the most crucial stage because it determines the outcome. The algorithm steps are given in Table 1.

Table 1. Algorithm steps of NN

1. Initialize weights $w$and biases$b$ of layer L, to determine theOccupancy Mapping
2. Calculate the neurons of output$outputneuron_j = \sum_{i=1-m} w_{ij}x_i + b_j$
3. Calculate data of output layers using$outputneuron_k = TargetValue_k + Error_K^L$
4. Compute the error E of neuron k$Error_j = outputneuron_j(1 - outputneuron_j)\sum_k Error_k w_{jk}$
5. Do steps 2 to 4 until network is congregated.

The general learning methods tend to converge too quickly and get stuck in local optima. Thus, training ANN with conventional learning methods is difficult because of the difficulty of achieving desired efficiency. The current work proposes an MDPO-based technique to learn the NN to address this issue (ONN). To find a NN weight vector with minimal RMSE, the MDPO is used.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_o - y_i)^2}$$

Here,$y_o$is the actual observation rate at time i and $y_i$is the prophesiedrate at time i.

Modified Dolphin Partner Optimization based NN: Better neural network models can be created by employing this optimization on NN parameters like weight and bias. Dolphins rely on echolocation to navigate, locate prey, and communicate with one another. And they team up to track out their prey, too. To find a mate, a dolphin will swim about its neighbourhood and approach its neighbours. All of his collaborators and he form an identity team that is currently functioning as an online dynamic team. Once a dolphin has assembled his squad, he may share information with his teammates to determine where everyone is in terms of fitness and where the team might improve (i.e. optimal parameter for NN). The RMSE of each resolution is going to be assessed to determine its fitness, and the most fit solution will be chosen to serve as the NN variable. All of the NN parameters needed to fulfil user requests are initialised to 'j' in this algorithm. Any number between zero and five hundred can represent j. With this method, the MDPO chooses a collaborator and forms an autonomous, responsive team.

After assembling a team, each dolphin shares information with its associates to determine where everyone stands in terms of fitness and how best to proceed. Every one of these NNs is trained with a hyperparameter x, where x=0....n, where n is the greatest number of parameters that can be used in a NN. In their hunting territory, dolphins solely target schools of

sardines, which they either seek for using certain parameters or ambush in large numbers.

At first, dolphins look everywhere before zeroing down on their prey (exploration phase). They zero in on a smaller and smaller area to look in and ramp up their click rate as they draw closer to their prey (exploitation phase). The exploration and exploitation capabilities of the MDPO algorithm are governed by the fast convergence factor (FCF), meaning the proportion of the most often occurring response. In this algorithm, we modify the convergence factor according to a function that characterises the convergence rate. The following is a description of this function:

$$Pr(l_i) = Pr_1 + (1 - Pr_1) * \left(rand(s_i - 1) + rand(Ns - 1)\right)$$

where $Pr(l_i)$ is a probability at the i-th iteration, Pr_1 is a probability at the i-th iteration, s i is the iteration quantity, and Ns is the quantity of iterations required to find the best solution. Repeatedly updating the dolphin's position in this manner will be done until either the target number of iterations is reached or the maximum maximal parameter for NN is discovered. Changing hyperparameter values using MDPO's pseudocode is outlined in Table 1. Flowchart of DPO based hyperparameter value Selection is shown in fig 2.

Table1.The pseudocode of MDPO based hyperparameter value Selection

Input: Number of prey (parameters of NN) on swarms of sardines in their search space n, parameters values j
Output: Optimal value of parameters
Initialize total number of n
Do
For each n
Calculate fitness Fit using RMSE formula of each location
if$\left(F_i > F_j\right)$
Calculate FCF
Select the options of each variable in next iteration through Pr
Calculating the next positions according to the probability assigned to each option
Replace the j by new solutions
End
Update the solution and return optmal valuesof NN
End for

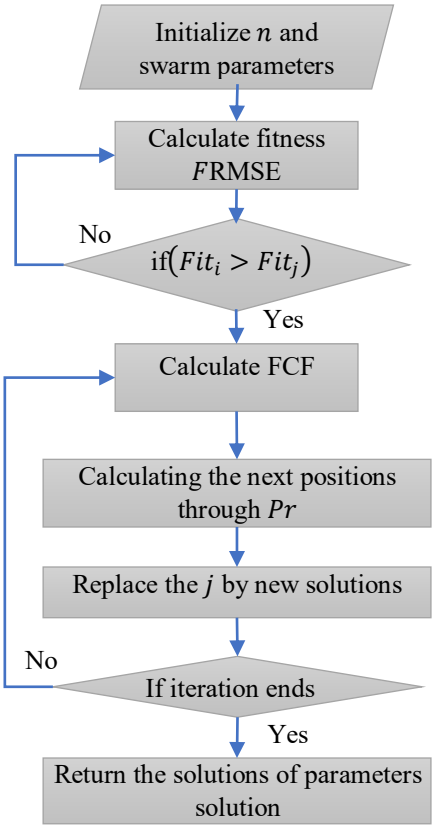

Fig.2.Flowchart of DPO based hyperparameter value Selection

IV. RESULT AND ANALYSIS

By implementing three different algorithms Kalman Pose Estimation Filter and Rao Blackwellized Particle Filter and OGM algorithm in the dataset is shown in Fig 2 and Fig 3, we can predict the OGMs. On comparing the values of measurement data with ground truth and the estimated readings correctly predicts occupancy grid mapping of local environment.

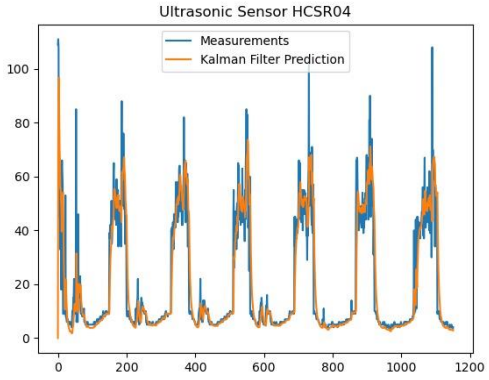

Fig 3: Optimal Estimation of Sonar Data with Kalman Filter

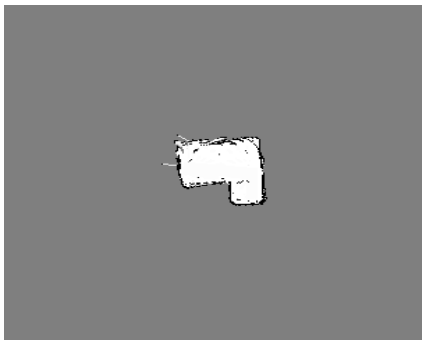

Fig 4: Prediction Map from Estimated Sonar Feed

V. CONCLUSION AND FUTURE SCOPE

The back-propagation model is trained the NN using the sensory values. The results of the studies using the sensing system and the map-building technique produced extremely respectable outcomes. To improve the accuracy of sensor readings, sensor fusion techniques and filter techniques for pre-processing specified data employing RBPF and EKF are used. These sensor data are useful in the map-building process since they are assessed using actual data collected in urban settings. Let's assume that a similar device operated in the future by a mobile robot and sensing system will generate more precise sensory values with location estimation.

REFERENCES

[1] Veeraswamy, A., Galea, E. R., Filippidis, L., Lawrence, P. J., Haasanen, S., Gazzard, R. J., & Smith, T. E. (2018). The simulation of urban-scale evacuation scenarios with application to the Swinley forest fire. Safety science, 102, 178-193.

[2] Kim, C. H., Lee, J. Y., & Lee, J. J. (2003). Feature extraction method for a robot map using neural networks. Artificial Life and Robotics, 7(3), 86-90.

[3] Meyer-Delius, D., Beinhofer, M., & Burgard, W. (2012, July). Occupancy grid models for robot mapping in changing environments. In Twenty-Sixth AAAI Conference on Artificial Intelligence.

[4] Fainberg, J., Renals, S., & Bell, P. (2017, August). Factorised Representations for Neural Network Adaptation to Diverse Acoustic Environments. In INTERSPEECH (pp. 749-753).

[5] Schreiber, M., Belagiannis, V., Gläser, C., & Dietmayer, K. (2020, May). Motion estimation in occupancy grid maps in stationary settings using recurrent neural networks. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 8587-8593). IEEE.

[6] Koenker, R., & Mizera, I. (2014). Convex optimization, shape constraints, compound decisions, and empirical Bayes rules. Journal of the American Statistical Association, 109(506), 674-685.

[7] Sreekanth, K. V., Alapan, Y., ElKabbash, M., Wen, A. M., Ilker, E., Hinczewski, M., ... & Strangi, G. (2016). Enhancing the angular sensitivity of plasmonic sensors using hyperbolic meta materials. Advanced optical materials, 4(11), 1767-1772.

[8] Taneja, S., Akinci, B., Garrett Jr, J. H., & Soibelman, L. (2016). Algorithms for automated generation of navigation models from building information models to support indoor map-matching. Automation in Construction, 61, 24-41.

[9] Danescu, R., Oniga, F., & Nedevschi, S. (2011). Modeling and tracking the driving environment with a particle-based occupancy grid. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1331-1342.

[10] Zhao, C., & Sander, H. A. (2015). Quantifying and mapping the supply of and demand for carbon storage and sequestration service from urban trees. PLoS One, 10(8), e0136392.

[11] Zhao, H., & Wang, Z. (2011). Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion. IEEE Sensors Journal, 12(5), 943-953.

[12] Nagla, K. S., Uddin, M., & Singh, D. (2012). Improved occupancy grid mapping in specular environment. Robotics and autonomous Systems, 60(10), 1245-1252.

[13] Ahmed, M. A., Zaidan, B. B., Zaidan, A. A., Salih, M. M., Al-Qaysi, Z. T., & Alamoodi, A. H. (2021). Based on wearable sensory device in 3D-printed humanoid: A new real-time sign language recognition system. Measurement, 168, 108431

[14] Koenka, I. J., Sáiz, J., & Hauser, P. C. (2014). Instrumentino: An open-source modular Python framework for controlling Arduino based experimental instruments. Computer Physics Communications, 185(10), 2724-2729.

[15] Kim, K. J., Pun, M. O., & Iltis, R. A. (2010). Joint carrier frequency offset and channel estimation for uplink MIMO-OFDMA systems using parallel Schmidt Rao-Blackwellized particle filters. IEEE Transactions on Communications, 58(9), 2697-2708.

[16] G. Dency Flora, G. Sekar, R. Nivetha, R. Thirukkumaran, D. Silambarasan and V. Jeevanantham, "An Optimized Neural Network for Content Based Image Retrieval in Medical Applications," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022, pp. 1560-1563, doi: 10.1109/ICACCS54159.2022.9785151.

[17] Haldorai, A., Ramu, A Canonical Correlation Analysis Based Hyper Basis Feed forward Neural Network Classification for Urban Sustainability (2021) Neural Processing Letters, 53 (4), pp. 2385-2401

[18] Jyothsna, C., Srinivas, K., Bhargavi, B., Sravanth, A. E., Kumar, A. T., & Kumar, J. S. (2022, May). Health Insurance Premium Prediction using XGboost Regressor. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 1645-1652). IEEE.