Autonomous utility tunnel inspection robot

Abijith P Mohan[1], Alexander Philip[2], Aswin s pillai[3], Vishnuraj A[4], Prof. Beena M Varghese[5], Prof. Linu Jose[6]
UG Student, Professor[5], Professor[6]
Department of Electrical and Electronics Engineering
Mar Athanasius College of Engineering,Kerala,aswinspillai54@gmail.com

***Abstract*— Autonomous utility tunnel inspection robots require precise navigation and robust movement prediction based on rigorous theoretical principles. Central to a robot navigation system is reliable mapping, which is crucial for normal operations and adaptation to dynamic environments where moving objects alter the landscape. This is an approach to enhance mapping for autonomous robots using 3D Point Cloud Mapping . The generated point cloud is invaluable for detecting leaks or cracks within tunnels, a task that is challenging to accomplish manually. The autonomous system leverages the Robotic Operating System (ROS) for software integration and uses Extended Kalman Filter as a part of sensor fusion. Captured data is processed and delivered to mine inspection officers for analysis and decision-making. Additionally, the high-resolution mapping capabilities of the 3D point cloud facilitate detailed inspection and maintenance planning, thereby improving safety and operational efficiency in tunnel environments. This approach not only automates the inspection process but also ensures consistent and accurate detection of structural issues, contributing to the overall reliability of tunnel infrastructure.**

***Index Terms*—A* path planning, Extended Kalman filter, Forward Kinematics, Inverse Kinematics**

I. INTRODUCTION

Autonomous tunnel inspection robot navigate autonomously within the civil infrastructures grab images and analyse them in order to identify defect types.The realm of infrastructure maintenance has witnessed a remarkable evolution, with the advent of autonomous utility tunnel inspection robots.These advanced machines, equipped with sophisticated sensors and other devices, are revolutionizing the way one,assesses and maintain underground infrastructure. By automating the often hazardous and time consuming task of tunnel inspection, these robots are enhancing safety, efficiency, and accuracy. Autonomous utility tunnel inspection robots are designed to navigate through confined spaces and complex terrains. They are typically equipped with cameras, sensors, and other diagnostic tools that enable them to gather data on the tunnel's structural integrity, corrosion, leaks, and other potential problems. The robots can operate autonomously or with remote human supervision, allowing for efficient and safe inspections.The methodology include manual control,Mapping[1],Sensorfusion[2], localization[2],Path planning.Mapping is done using RTAB Algorithm[4],which is a graph based slam approach that uses data collected from vision sensors to localize the robot and map the environment.Sensor fusion is the process of combining data from multiple sensors to create more accurate and reliable representation of a robot's state or environment. Localization is the process of determining a robot's position and orientation within a known environment.Path planning refers to the process of generating a route for a robot to move from its current position to a target destination while destination while avoiding obstacles and considering environmental constraints and is done using A* path planning method, Mathematical modelling include Forward kinematics, Inverse kinematics,
Deadreckoning, Extended kalman filter. Hardware components used are Nvidia Jetson orin developer kit which empowers the development of AI powered robots,Arduino mega,Buck converter,cytron dual motor driver,Encoder motor,Intel real sense d435i camera.Software used is Robot operating system(ROS) which is a set of software libraries and tools that helps to build robot applications.

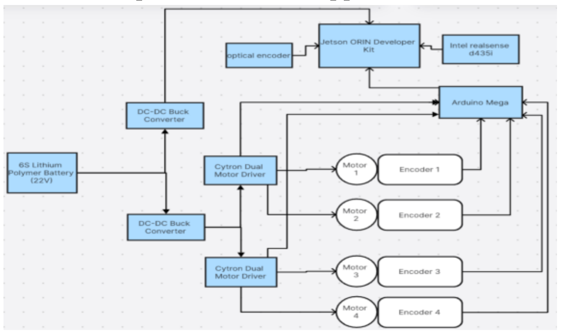
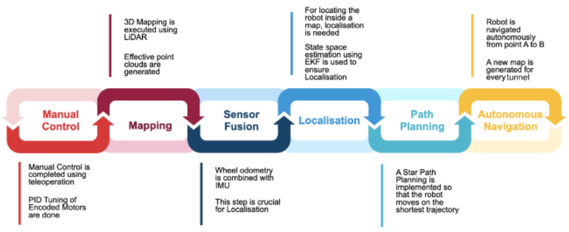


Fig.1 shows block diagram

It inludes a 6S Lithium polymer battery which supplies a 22 V supply to dc-dc buck converter .Buck converter steps down the voltage and is supplied to Jetson orin develepor kit and cytron dual motor driver.Motor driver is connected to Encoder motor and Arduino mega.Arduino mega is further connected to Jetson orin.Intel realsense d435i camera and optical encoder is also connected to Jetson orin.


3D Mapping is executed using LiDAR
Effective point clouds are generated
For locating the robot inside a map, localisation is needed
State space estimation using EKF is used to ensure Localisation
Robot is navigated autonomously from point A to B
A new map is generated for every tunnel
Manual Control
Mapping
Sensor Fusion
Localisation
Path Planning
Autonomous Navigation
Manual Control is completed using teleoperation
PID Tuning of Encoded Motors are done
Wheel odometry is combined with IMU
This step is crucial for Localisation
A Star Path Planning is implemented so that the robot moves on the shortest trajectory

The methodology includes manual tuning which is done using teleop twist which converts keyboard inputs into twist messages which corresponds to directional commands allowing the user to navigate rover intuitively,Mapping is done using RTAB mapping algorithm,which is a SLAM approach designed for mapping.RTAB is used to construct 3D maps of environments using sensors like RGB-D cameras,Lidar .Sensor fusion is the process in which odom values is combined with IMU data,Localization is done using Extended Kalman filter. Extended kalman filter which uses knowledge of physics of motion to make small adjustments to actual measurement to reduce amount of noise to get a better estimate.Path planning is done using A* path planning.A* star is a path finding algorithm that's finds shortest path between initial and final destination.Mathemathical modelling inludes forward kinematics which gives final position of the robot,Inverse Kinematics helps to calculate velocity.

II. FORWARD KINEMATICS

Forward kinematics is used to determine the final position of the robot using its initial position and velocity. Here , $x_C{}^0$ is the displacement along x axis, $y_c{}^0$ is the displacement along y axis, γ is the orientation of body center, $x_c{}^1$ and $y_c{}^1$ along new axis,Wb is the wheel base.

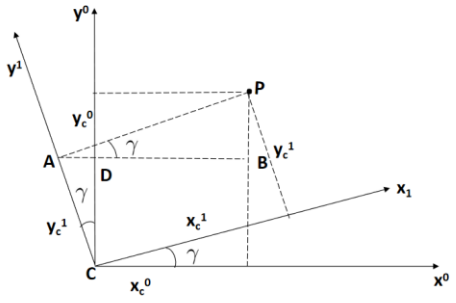


Fig.2 Shows the initial and final position of the robot

$$X_c^0 = X_c^1 \cos\gamma - Y_c{}^1 \sin\gamma \tag{1}$$

$$Y_c^0 = Y_c^1 \sin\gamma + Y_c{}^1 \cos\gamma \tag{2}$$

Matrix form representation of the above equation,

$$\begin{bmatrix} X_c{}^0 \\ Y_c{}^0 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} X_c^1 \\ Y_c^1 \end{bmatrix} \tag{3}$$

It can be further be return as,

$C^0 = R_1{}^0 C^1$

Assume that robot is moving straight,

$\dot{\phi}_L$ and $\dot{\phi}_R$ are non zero and r is the radius

$$\dot{X}_c = \left(r\dot{\phi}_L\right)/2 + \left(r\dot{\phi}_R\right)/2 \tag{4}$$

$$\dot{Y}_c = 0 \tag{5}$$

$$\dot{X}_c = (r/2)\cos\gamma\left(\dot{\phi}_L + \dot{\phi}_R\right) \tag{6}$$

$$\dot{Y}_c = (r/2)\sin\gamma\left(\dot{\phi}_L + \dot{\phi}_R\right) \tag{7}$$

To find the orientation of robot,

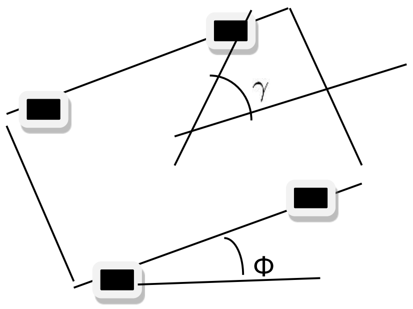


Fig.3 Shows the orientation of the robot

Consider an arc with radius r,

$$wb\gamma = r\phi$$

Differentiating,

$$wb\dot{\gamma} = r\dot{\phi}$$

$$\dot{\gamma} = (r\dot{\phi})/wb$$

To find orientation of robot consider three cases,

$$\text{Case 1: } \dot{\phi}_L = 0, \dot{\phi}_R \text{ non zero}$$

$$\dot{\gamma} = \left(r\dot{\phi}_R\right)/wb \quad \text{(anticlockwise)} \tag{8}$$

$$\text{Case 2: } \dot{\phi}_R = 0, \dot{\phi}_L \text{ non zero}$$

$$\dot{\gamma} = \left(-r\dot{\phi}_L\right)/wb \quad \text{(clockwise)} \tag{9}$$

$$\text{Case 3: straight,}$$

$$\dot{\gamma} = (r/wb)\left(\dot{\phi}_L - \dot{\phi}_R\right) \tag{10}$$

$$\begin{pmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} (r/2)\cos\gamma\left(\dot{\phi}_L + \dot{\phi}_R\right) \\ (r/2)\sin\gamma\left(\dot{\phi}_L + \dot{\phi}_R\right) \\ (r/wb)\left(\dot{\phi}_R - \dot{\phi}_L\right) \end{pmatrix} \tag{11}$$

On integrating the above eq.(11),position and orientation can be obtained.

III. INVERSE KINEMATICS

Inverse kinematics is used to calculate the velocity using initial and final position.

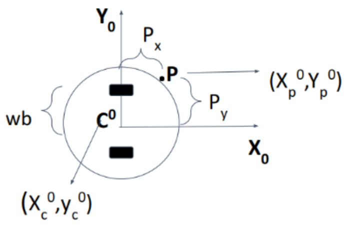


Fig.4 shows the initial position of robot

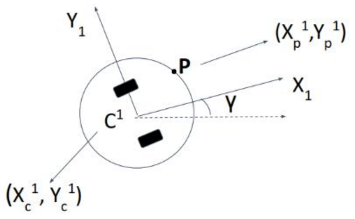


Fig. 5 shows the final position of robot

From forward kinematics,

$C^0=R_1^0C^1$, $P^0=R_1^0P^1$, $P^0-C^0 = R_1^o(P^1-C^1)$

$$\begin{pmatrix} x_p^0 - x_c^0 \\ y_p^0 - y_c^0 \end{pmatrix} = \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} x_p^1 - x_c^1 \\ {y_p}^1 - {y_c}^1 \end{pmatrix} \tag{12}$$

$$\begin{pmatrix} X_p^0 \\ Y_p^0 \end{pmatrix} = \begin{pmatrix} X_c^0 \\ Y_c^0 \end{pmatrix} + \begin{pmatrix} \cos\gamma - \sin\gamma \\ \sin\gamma\cos\gamma \end{pmatrix} \begin{pmatrix} P_x \\ P_y \end{pmatrix} \tag{13}$$

$$\begin{pmatrix} \dot{x}_p^0 \\ \dot{y}_p^0 \end{pmatrix} = \begin{pmatrix} \dot{x}_c^0 \\ \dot{y_c}^0 \end{pmatrix} + \begin{pmatrix} (-\sin\gamma)\dot{\gamma} & (-\cos\gamma)\dot{\gamma} \\ (\cos\gamma)\dot{\gamma} & (-\sin\gamma)\dot{\gamma} \end{pmatrix} \begin{pmatrix} P_x \\ P_y \end{pmatrix} \tag{14}$$

From forward kinematics, position and orientation is obtained which can be written as,

$$\dot{X}_c^0 = v\cos\theta, \dot{Y}_c^0 = v\sin\theta, \dot{\theta} = \omega$$

$$\begin{pmatrix} \dot{X}_p^0 \\ \dot{Y}_p^0 \end{pmatrix} = \begin{pmatrix} \cos\gamma & -P_x\sin\gamma - P_y\cos\gamma \\ \sin\gamma & P_x\cos\gamma - P_y\sin\gamma \end{pmatrix} \begin{pmatrix} V \\ \omega \end{pmatrix} \tag{15}$$

$$\dot{X}_p^0 = Kp_x\left(X_{\text{ref}}(\text{t}) - X_{\text{p}}^0\right) \text{ and } \dot{Y}_{\text{p}}^0 \quad = Kp_y\left(Y_{\text{ref}}(\text{t}) - {Y_{\text{p}}}^0\right) \tag{16}$$

$$\begin{pmatrix} Kp_x\left(X_{\text{ref}} - X_y^0\right) \\ Kp_y\left(Y_{\text{ref}} - {Y_p}^0\right) \end{pmatrix} = \begin{pmatrix} \cos\gamma & -P_x\sin\gamma - P_y\cos\gamma \\ \sin\gamma & P_x\cos\gamma - P_y\sin\gamma \end{pmatrix} \begin{pmatrix} V \\ \omega \end{pmatrix}$$

$$\begin{pmatrix} V \\ \omega \end{pmatrix} = \begin{pmatrix} Kp_x\left(X_{\text{ref}} - {X_p}^0\right) \\ Kp_y\left(Y_{\text{ref}} - {Y_p}^0\right) \end{pmatrix} \begin{pmatrix} \cos\theta - (P_y/P_x)\sin\gamma & \sin\theta + (P_y/P_x)\cos\gamma \\ -(1/P_x)\sin\gamma & (1/P_x)\cos\gamma \end{pmatrix}$$

IV. DEAD RECKOING

Dead reckoning is a navigation technique that uses a known position and information about velocity or distance traveled to estimate an object's current position.It uses sensor readings from gyroscopes,accelerometers and magnetometers to estimate an object's position.

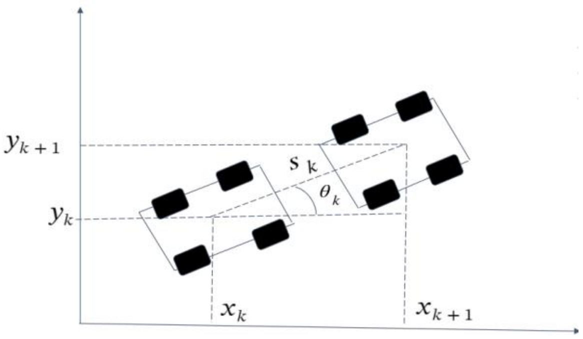


Fig.6 shows movement of robot

Final x position and y position is given by initial x position and y position plus displacement along x position and y position.

$$\hat{x}_{k+1} = \hat{x}_k + \tilde{s}_k\cos\theta_k \tag{17}$$

$$\hat{y}_{k+1} = \hat{y}_k + \tilde{s}_k\sin\theta_k \tag{18}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \Delta\tilde{\theta}_k \tag{19}$$

The above equation shows the final estimated x-position,y-position and orientation.

V. EXTENDED KALMAN FILTER (EKF)

Extended kalman filter is an improved version of kalman filter which is used to reduce errors from sensor measurements and helps in sensor fusion of odom and sensor measurements.

It include state estimation,state prediction,covariance prediction,kalman gain calculation and state updation.

Step 1: Initial conditions

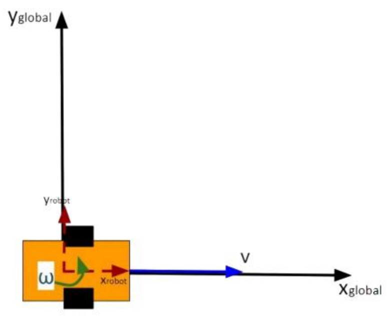


Fig. 7 shows the initial position of the robot

$$\hat{x}_{k-1|k-1} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \gamma_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (20)$$

$$\hat{x}_{k-1|k-1} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \gamma_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (21)$$

Initially the robot is at rest and orientation is zero

Step 2: Predicted state estimate

Here,robot's current is calculated using previous state and control inputs.

$$\begin{bmatrix} x_t \\ y_t \\ \gamma_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \gamma_{t-1} \end{bmatrix} + \begin{bmatrix} \cos\gamma_{t-1} * dt & 0 \\ \sin\gamma_{t-1} * dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v_{t-1} \\ \omega_{t-1} \end{bmatrix} + \begin{bmatrix} \text{noise}_{t-1} \\ \text{noise}_{t-1} \\ \text{noise}_{t-1} \end{bmatrix}$$

Step 3: Predicted covariance of state estimate.This shows the uncertainity in robot's state.

$$\text{Predicted covariance estimate} \quad \boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^{\top} + \boldsymbol{Q}_k \qquad (22)$$

Action uncertainity matrix Q $_K$ which is a 3*3 identity matrix gives uncertainity due to movement.It decide whether estimated value or measured value is considered.

Initial covariance matrix P $_{K-1|K-1}$ which shows the initial guess values.

F $_K$ is a 3*3 identity matrix which is similar to A matrix in state space equation Y =AX+BU.The transpose of F $_K$ is also an identity matrix.

$$P_{k|k-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The equation shows predicted covariance at time k.

Step 4: Innovation residual

Innovation residual is the difference between sensor actual measurement and predicted value.

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h\left(\hat{\boldsymbol{x}}_{k|k-1}\right) \qquad (23)$$

$$z_k = \begin{bmatrix} x_k \\ y_k \\ \gamma_k \end{bmatrix} \qquad H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

H $_K$ is the measurement matrix which is used to convert state estimate to sensor measurement.

Observation model is given by,

$$h\left(\hat{x}_{k|k-1}\right) = H_k \hat{x}_{k|k-1} + w_k$$

Step 5: Innovation covariance

It shows error in prediction as a matrix.Innovation covariance fiter out noise from innovation residual.

R $_K$ is the sensor measurement covariance matrix which shows noise in sensor measurement.

$$\text{Innovation (or residual) covariance} \quad \boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\top} + \boldsymbol{R}_k$$

Step 6: Kalman gain

$$\text{Near-optimal Kalman gain} \quad \boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^{\top} \boldsymbol{S}_k^{-1} \qquad (24)$$

If kalman gain is greater than zero,estimated value is considered and if kalman gain decreases to zero sensor value is considered.

Step 7: Updated state estimate and covariance of state

$$\text{Updated state estimate} \quad \hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k$$

$$\text{Updated covariance estimate} \quad \boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1}$$

COMPARISON OF DEAD RECKONING AND EKF

EKF is used in non linear systems while dead reckoning in linear system.EKF is more accurate while dead reckoning is less accurate.EKF is more complex while dead reckoning is simple to implement. EKF require external sensor data while Dead reckoning does not require external sensor data.

VI. SIMULATION

Motor simulation:

Simulation is done in MATLAB.Tuning methods include Genetic algorithm,Manual tuning,Using PID Function,using PID Tuner.From this,Manual tuning is used because of quick rise time and less settling time.

EKF:

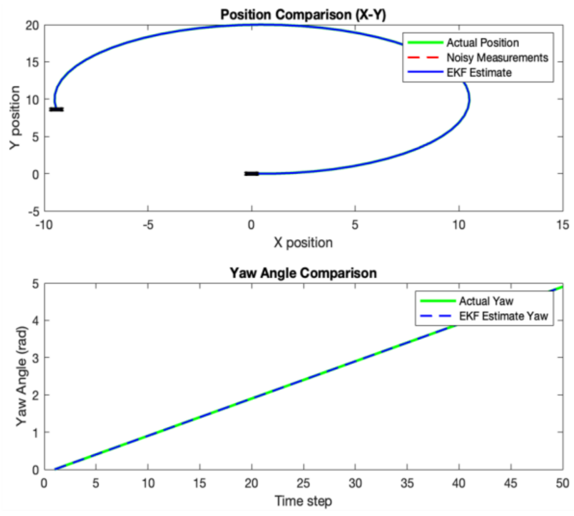


Fig.8 shows behavior of EKF in a noiseless system

Fig.8 shows the Extended kalman filter in a noiseless system.Graph shows the estimated value is equal to the sensor measurements.

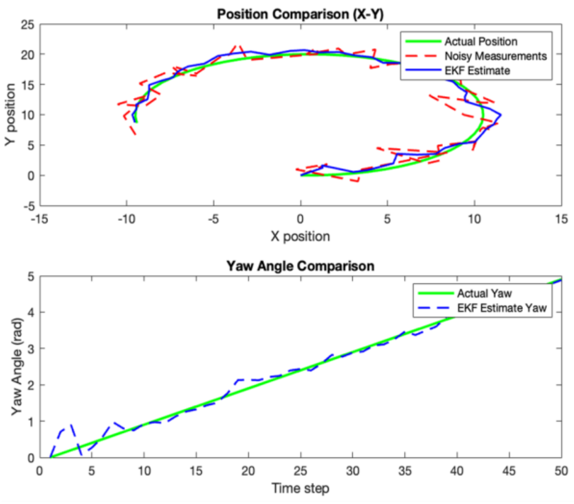


Fig .9 shows behavior of EKF in a noisy system

Fig.9 shows the behavior of EKF in a noisy system and the result shows estimated value is more accurate than sensor value.

VII. RTAB MAPPING

Robots need to understand their environment for safe and efficient navigation. Mapping helps robots localize themselves in a known or unknown space. Many mapping techniques struggle in large-scale or changing environments. RTAB-Map handles loop closures and memory management, making it ideal for long-term navigation. Some SLAM (Simultaneous Localization and Mapping) methods are computationally expensive. RTAB-Map efficiently manages real-time data processing and memory constraints. RTAB-Map combines data from RGB-D cameras, IMU, and odometry to improve accuracy. It is widely used in mobile robots, drones, and autonomous vehicles.

Working of RTAB Mapping:
Sensor inputs: Takes RGB-D camera(eg:Intel realsense,Kinect,ZED),Lidar,IMU, or stero vision as input.
Feature Extraction and Visual Odometry: Detects keypoints in images and tracks them across frames.Estimates the robot’s motion(pose transformation) using visual odometry(VO).
Loop Closure Detection: Recognizes when the robot revisits a previously mapped area.Uses Bag-of-words(BoW) for appearance based mapping.
Graphoptimization(PoseGraphSLAM):Maintain a graph of poses and corrects drift over time.Uses GTSAM or g2o for graph optimization.
Memory Management: Efficiently manages map storage by removing reduntant information.Allows robots to operate for long durations without excessive memory use.

VII. A* PATH PLANNING

A* (A-star) is a widely used pathfinding algorithm that finds the shortest route between a start point and a goal while considering obstacles. It’s a graph-based search algorithm that combines features of both Dijkstra’s Algorithm and the Greedy Best-First Search to efficiently find the optimal path.

A* pseudocode:

```
1.Initialise open list(starting node) and closed list(empty).
2.While the open list is not empty
  a) pick the node with lowest f(n) value.
  b) If it’s goal ,return path.
  c) Move it to the closed list.
  d) Generate the neighbor nodes.
```

e) If not in the closed list or a better path exist ,update g(n),h(n) and f(n).

3. If no path is found,return failure.

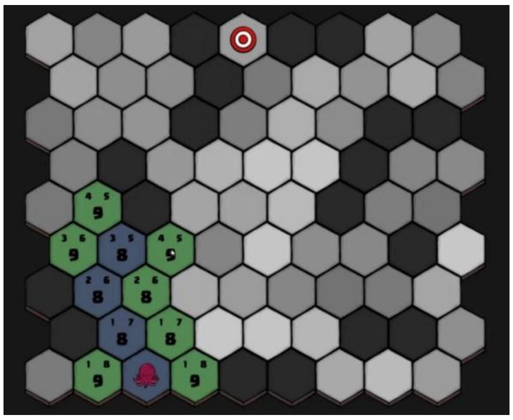


Fig.10 shows initial position

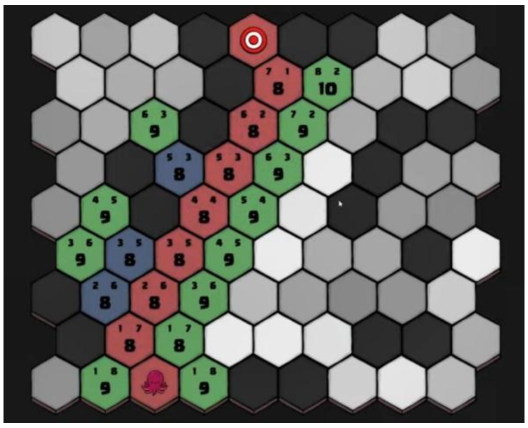


Fig.11 shows path followed to reach final position

VIII. SOFTWARE

ROS1:

ROS 1 (Robot Operating System 1) is an open-source robotics middleware designed to help developers build, simulate, and deploy robotic applications.It provides a structured framework for managing hardware communication, sensor integration, and motion planning in robots. It uses a publisher-subscriber model (topics) and client-server (services). Enables multiple nodes (processes) to communicate asynchronously. It supports LiDAR, cameras, IMUs, motors, grippers, etc and works with TurtleBot, UR robots, Clearpath, DJI drones, etc. Uses Gazebo to simulate robot behavior in 3D environments and RViz to visualise sensor data, maps, and robot states. ROS Master manages communication between ROS nodes. Nodes are independent processes that perform specific tasks.Topics are message channels for data exchange. Services involve request-response model for robot commands. Launch Files automate starting multiple ROS nodes.

EXPERIMENTAL SETUP

The main components include 6S lithium polymer battery,DC-DC Buck converter,Jetson Orin develepor kit,Cytron dual motor driver Encoder motor,Arduino mega,Intel realsense d435i camera,optical encoder.

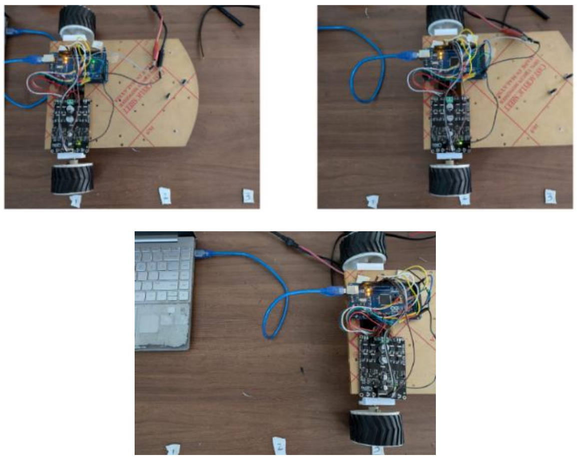


Fig.12 shows Drive mechanism

Hardware implementation of encoded dc servo motor is carried out.Four encoded dc servo motors are used,which is connected to cytron dual motor driver and Arduino mega.Encoder determines the rotational speed of the motor.Tunnel inspection robot utilizes DC servomotors with encoders,cytron dual motor drivers to achieve accurate position and speed control,which is critical for maneuvering in confined spaces. Encoders provide precise feedback on angular displacement, enabling accurate velocity calculations. Tunnel inspection robots require high torque to move through uneven terrain and climb slopes,so the chosen motor has a torque of 25.1 Kgcm.The low speed motor is chosen with speed 100 rpm.PWM is used to control the speed and direction of motors in tunnel inspection robots.PWM enables precise speed control allowing smooth and accurate movements.The range of PWM is about 0 to 255.so,the tunnel inspection robot has a minimum range of 120 and maximum of 190.so,the robot moves with a speed of 40-75 percentage.The velocity is derived using the tick values from encoders, experimentally calibrated at 2100 ticks per meter, and time measurements via the millis() function, which compensates for delays and minor errors.

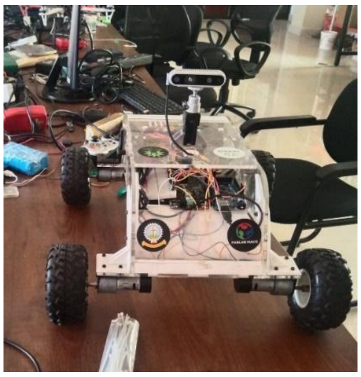


Fig.13 shows robot model

YOLOv8 OBB

YOLOv8 OBB (Oriented Bounding Box) is an extension of the YOLOv8 object detection model designed to detect objects with rotated bounding boxes instead of traditional axis-aligned boxes. This is particularly useful for applications like aerial imagery, document analysis, and industrial inspections, where objects may not always be aligned with the image axes.The YOLOv8 OBB model follows the standard YOLO detection pipeline but modifies the output to predict oriented bounding boxes.Oriented Bounding Boxes (OBB) include an additional angle to enhance object localization accuracy in images. Unlike regular bounding boxes, which are axis-aligned rectangles, OBBs can rotate to fit the orientation of the object better. This is particularly useful for applications requiring precise object placement, such as aerial or satellite imagery.OBB improves detection accuracy for objects with arbitrary orientations, especially in tasks like ship detection, text recognition, and traffic sign identification.

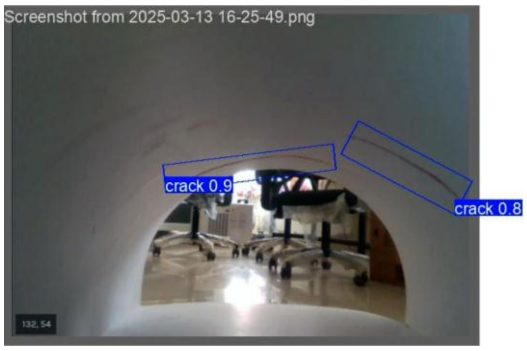


Fig.14 shows Final Machine learning model

Fig 14.. provides a close-up view of detected cracks, the model identifies and labels cracks in a curved surface, assigning confidence scores (e.g., 0.9 and 0.8) to its predictions. A confidence score in machine learning, particularly in object detection models like YOLOv8, represents the model's certainty that a detected object (e.g., a crack) is correctly identified. It is a probability value between 0 and 1, where higher scores (e.g., 0.9 or 0.8) indicate stronger confidence that the detected object is indeed what the model predicts it to be and lower scores (e.g., 0.3 or 0.4) suggest lower confidence, meaning the detection might be uncertain or incorrect.

CONCLUSION

Autonomous Utility Tunnel Inspection Robot developed a foundational design and control approach for autonomous utility tunnel inspection robot, leveraging concepts such as forward kinematics, inverse kinematics, extended kalman filtering, and PID tuning. These principles contributed to creating a robot capable of precise navigation and effective data collection within the complex and confined environment of utility tunnels.The implementation of forward and inverse kinematics ensured that the robot's movement and positioning could be accurately calculated and controlled, which is essential in narrow tunnel pathways. Extended kalman filtering allowed the the system to maintain accurate localisation and pose estimation, even when faced with sensor noise and minor environmental uncertainties. Fine-tuning the PID controllers resulted in smoother and more responsive control, leading to safer and more efficient operations. The simulations demonstrate the effectiveness of various PID tuning methods, with the genetic algorithm providing the quickest response due to zero settling time, followed by manual tuning, which shows the shortest rise time. The Extended Kalman Filter (EKF) proves highly effective for both position and yaw estimation, maintaining accuracy in noiseless environments and significantly reducing noise effects in realistic scenarios. These results highlight the EKF’s robustness in enhancing sensor-based navigation, establishing its critical role in achieving precise, stable control for autonomous systems under varied environmental conditions.Hardware has been completed and an appropriate motor has been selected having a PWM range of 120 to 190.Manual control is achieved through UART communication protocol.

REFERENCES

[1] Tian, H. Liu, Z. Liu, H. Li, and Y. Wang, “Research on multi-sensor fusion slam algorithm based on improved gmapping,” IEEE Access, vol. PP, 01 2023.

[2] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, “Robust and precise

vehicle localization based on multi-sensor fusion in diverse city scenes,”pp. 4670–4677, 05 2018.

[3] A. Eman and R. Hedjar, “Mobile robot localization using extended kalman filter,”pp. 1–5, 03 2020.

[4] L.-B. Chen, X.-R. Huang, and W.-H. Chen, “Design and implementation of an artificial intelligence of thingsbased autonomous mobile robot system for pitaya harvesting,” IEEE Sensors Journal, vol. 23, pp. 13220–13235, 06 2023.

[5] G. Zhang, Z. Zhisheng, X. Zhijie, D. Min, P. Meng and J. Cen, “Implementation and research on indoor mobile robot mapping and navigation based on RTAB-Map” IEEE Conf. Mechatronics, 2022.

[6] J. Prada, M. Luna, S. Park, and C. Song, “A deep learning technique as a sensor fusion for enhancing the position in a virtual reality micro-environment,” pp. 4935–4940, 10 2022.

[7] S.-S. Lee, M.-G. Choi, J. Park, and J.-M. Lee, “Localization of a high-speed mobile robot using ultrasonic rf sensor and global features,” Journal of Institute of Control, Robotics and Systems, vol. 15, pp. 734–741, 07 2009.

[8] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India,2024.

[9] Sangale, V. Shendre,“Localisation of a Mobile Autonomous Robot Using Extended Kalman Filter”IEEE Conf. 2013.

[10] G. Wan, X. Yang, R. Cai & S. Song, “ Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes” Proc. IEEE Int. Conf. Robot. Autom. May 2018.

[11] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006, doi: 10.1109/MRA.2006.1638022.

[12] H. T. Tran *et al.*, "Extended Kalman Filter (EKF) Based Localization Algorithms for Mobile Robots Utilizing Vision and Odometry," 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy, 2022, pp. 91-96, doi: 10.1109/MELECON53508.2022.9843066.