

DEVELOPMENT OF AN ADVANCED AUTONOMOUS VEHICLE CONTROLLER SYSTEM USING NONLINEAR MODEL PREDICTIVE CONTROL (NMPC) AND LONG SHORT-TERM MEMORY (LSTM)

Ogbuikwu Rowland I.¹, Raheem Rasheed Adebayo²

^{1,2} Department of Electrical and Electronic Engineering, Federal Polytechnic Ohodo, Enugu State, Nigeria

Email: rowlandikechukwu@gmail.com¹, raheemrasheed2r@yahoo.com²

D.O.I: 10.5281/zenodo.18863364

Received: 02.12.2025 | Accepted: 29.12.2025 | Published: 20.01.2026

ABSTRACT

RESEARCH ARTICLE

This study presents the implementation of an advanced vehicle controller-localization system that integrates Nonlinear Model Predictive Control (NMPC) with a Long Short-Term Memory (LSTM) network to improve autonomous vehicle path tracking performance. The proposed system utilizes sensor fusion approach of Visual-Inertial Odometry (VIO) and wheel odometry data through an Extended Kalman Filter (EKF) to provide accurate Three-Degree-Of-Freedom (3-DOF) for vehicle pose estimation in real time. The work was further implemented within a Robot Operating System (ROS)-based CARLA simulation environment and the LSTM model predicts future vehicle states over a finite horizon. Experimental results from the implementation show that the LSTM achieved low training and validation losses of 0.0148 and 0.0176, respectively and Mean Absolute Errors (MAE) of 0.08ms in position and 0.04 radians in heading prediction. Furthermore, the integrated NMPC-LSTM controller demonstrated an 18.6% reduction in trajectory tracking error compared to the baseline NMPC along with improved robustness under dynamic and noisy conditions. These findings validate the effectiveness of combining deep learning with model predictive control for enhancing autonomous vehicle navigation

KEYWORDS: Autonomous Vehicles; Nonlinear Model Predictive Control (NMPC); Long Short-Term Memory (LSTM); Visual-Inertial Odometry (VIO); Extended Kalman Filter (EKF)

INTRODUCTION

The development of electrified autonomous vehicles has gained attention recently. The most crucial move that an autonomous car has to be able to do is path tracking. An autonomous car must have a few key components in order to do this. The first essential component of the autonomous car is the controller. An autonomous ground vehicle requires a strong, quick, and steady controller to execute difficult path-tracking movements. However, the existence of non-holonomic limitations makes it difficult to build controls for stabilising such vehicles (Maghenem et al., 2021).

Second, the car should have a high degree of precision in identifying its orientation and position in the surroundings. Accurately estimating the location and orientation of the vehicle in various surroundings, which may include a range of weather conditions, is a requirement for sensors and state estimation algorithms. However, ground vehicle localisation has never been an easy task. According to Barzegar et al. (2020), most current vehicle localisation systems have receivers that can receive location and timing signals from at least one Global Navigation Satellite System (GNSS) element. Ground vehicles go through a range of various settings, such as highways, tunnels, urban canyons, woodland regions, and covered parking lots, in contrast to aerial vehicles, which often have a clear view of the sky and can readily receive location signals from the GNSS components. The GNSS is an unreliable positioning system for ground vehicles since its positioning signals are not available in all settings. But in recent years, scientists from all around the world have worked hard to address this issue.

Creating a suitable controller is the first obstacle in putting an autonomous car into operation. To enhance the effectiveness of path-tracking by the controllers in automobiles, researchers have worked hard. To control autonomous cars, several research have used a range of control strategies. The Proportional-Integral-Differential (PID) controller approach is one of the popular model-free controllers that have been employed in previous research (Zhao et al., 2012). Nevertheless, PID controllers are not appropriate methods when the system parameters vary over a large range (Barzegar et al., 2014). The tuning of this kind of controller is the second issue. Some academics have suggested artificial intelligence techniques for controller auto-tuning in an effort to mitigate the latter issue. However, under difficult driving situations, the controller still lacks enough stability (Alouche and Wu, 2018; Abdelhakim and Abdelouahab, 2019).

Another popular approach that has been utilised in several research is the Stanley technique (Thrun et al., 2007) (Amer et al., 2018; Dominguez et al., 2016). The longitudinal motion control of the vehicle was not considered in the initial design of this system, which was intended for lateral motion control. Even though the technique enhances the lateral controller's performance, it is still unable to take into account dynamic features like forces, which often manifest at high speeds. Another popular control strategy for ground vehicles is the Pure Pursuit controller (Morales et al., 2009). The controller was first presented as a way for ground robots to follow their paths. The method modifies the steering commands by using the look-ahead distance, which is the closest forward point on the path relative to the current position. But there are some issues with the controller.

In this work, we developed and put into practice a controller architecture, localisation algorithm, and intelligent vehicle communication system. Wheel odometry and visual inertial odometry (VIO) readings are combined with an extended Kalman filter to determine the system's complete state. We suggested a Nonlinear Model Predictive Control (NMPC) enhanced with the Long Short-Term Memory (LSTM) algorithm for the high-level control portion.

Proposed Method

The proposed method comprises of an NMPC-based route tracking module enhanced with an LSTM algorithm that predicts future states and control inputs ten steps in advance. Figure 1 displays the whole system design.

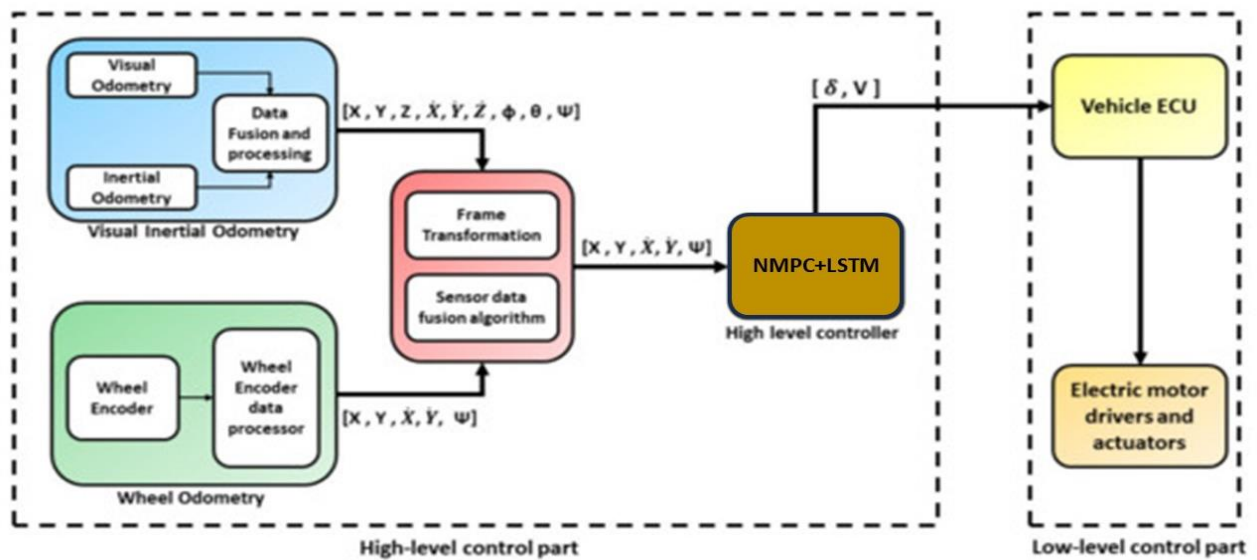


Figure 1: Architecture of the Controller Localization System

In this work, we fused odometry data from many techniques using an extended Kalman filter (EKF) to achieve precise state estimation. Data from the Visual Odometry (VO) algorithm and the Inertial Measurement Unit (IMU) are combined in the Visual-Inertial Odometry (VIO) algorithm. The VIO's output is the vehicle's three-dimensional (3D) location and orientation data. A wheel odometry processor unit, which is composed of a data processor and rotary encoders, simultaneously generates the vehicle's position and orientation data in two dimensions (2D). The final position and orientation estimator, which is an enhanced Kalman filter in the robot localisation package, receives the data from the two sources in the next phase. The final localisation unit does both sensor fusion and frame transformation. High-accuracy location and orientation data for the ground vehicle are provided by the unit. The NMPC+LSTM algorithm uses the produced data as input signals. The controller creates control directives that are transmitted to the vehicle's Electronic Control Unit (ECU)'s low-level control.

The Proposed Vehicle State Estimation, Localization and Sensor Algorithms

This study's localisation technique is based on combining data from wheel odometry with visual-inertial odometry. With reference to the car's starting position in the odometry frame, data from the previously stated sources is fed into an estimator algorithm that employs an extended Kalman filter to offer an almost precise estimate of the vehicle's location and heading. To improve the precision of vehicle trajectory estimate, Visual Inertial Odometry (VIO) uses sensor fusion between inertial measurement unit (IMU) measurements and camera image data (Kanellakis and Nikolakopoulos, 2017; Santoso et al., 2017). The architecture of the VIO system is summed up in Figure 2, where the front-end and back-end calculations are made to take use of both sensors, generate accurate ego-motion estimate, and accomplish strong performance.

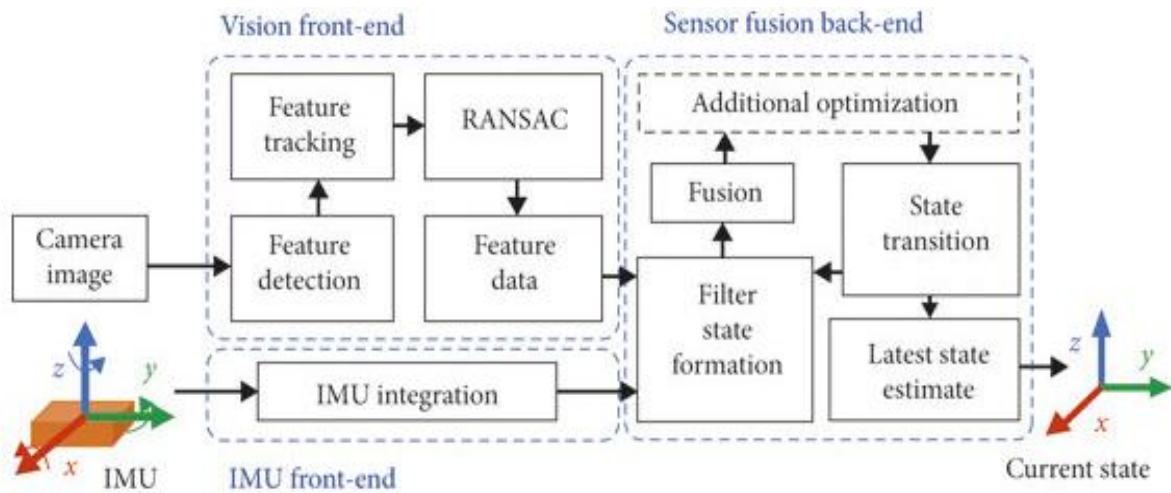


Figure 2: Overview of visual inertial odometry architecture (Nguyen et al., 2020)

Through various camera views, the vision front end computation aims to monitor 3D feature points. To fuse with the IMU data, geometric restrictions between various camera postures are developed (Mourikis and Roumeliotis, 2007; Nguyen et al., 2019; Wong and Majji, 2018). The sensor fusion approach determines the process's efficacy and estimation accuracy. Numerous sensor fusion systems with differing hardware computer resource needs have been presented in recent literature (Santoso et al., 2017). Nevertheless, VIO systems are hampered by the unavoidable buildup of errors. This restriction causes the system to progressively deviate and eventually stop tracking the trajectory of the vehicle after extended use. Only in short-term operation and short-distance travel does VIO generate a trustworthy estimate of the vehicle trajectory.

In this study, a Kalman-filter-based stereo visual inertial odometry is used. Data from the IMU and a stereo camera are used in the algorithm which describes the orientation, velocity and position of a moving automated vehicle with respect to the environment. The orientation is represented using quaternions, which are mathematical tools that avoid singularities common in other rotation representations like Euler angles. The velocity and position define how fast and where the object is moving in the environment and the model includes the biases of the gyroscope and accelerometer, which are small systematic errors present in sensor readings which must be estimated and corrected to improve accuracy. The IMU is often integrated with a camera, and their relative orientation and position must be known.

When multiple sensors like cameras are involved, the overall system includes not just the IMU state but also the states of each camera pose at different times which consist of their position and orientation errors. Together, the IMU and camera error states form the complete state error used in filtering and estimation algorithms. To estimate the evolving uncertainty of the system's state, a state-transition model is used to describe how the uncertainty grows over time as the system moves, and it uses process noise to account for sensor inaccuracies. To ensure that the uncertainties in the estimated position of the 3D feature don't distort the update process, the residual is projected into a subspace where the influence of this uncertainty is eliminated which allows the estimation filter, such as the Extended Kalman Filter (EKF), to remain consistent and reliable.

Finally, it's important to note that using a basic EKF for visual-inertial odometry can sometimes lead to errors in heading (direction estimation) which arises because the linearization used during prediction and measurement updates happens at slightly different points in time. To address this, improved filtering techniques such as the Observability-Constrained EKF (OC-EKF) (Huang et al., 2010), First-Estimate Jacobian EKF (FEJ-EKF) (Hesch et al., 2012), or robocentric mapping filters (Casyellanos et al., 2007) are used to maintain filter consistency over time by ensuring that important unobservable directions are treated correctly during estimation.

Wheel Odometry Algorithm

The information from encoders attached to the back wheels is used by the wheel odometry. For one tyre revolution, each encoder produces 100 sets of pulses (encoder resolution). One revolution of the tire causes the encoder to make one revolution (1 by 1 coupling). A minimum of two output signals (often A and B) in the form of digital square waves are provided by each rotary incremental encoder. The number of pulses indicates the covered distance, whilst the frequency of recurrence in the signal indicates the shaft speed rotation. Signals from encoders are sent to a processing board. Every five milliseconds, the encoder's signal is sampled by the CPU. The vehicle direction Ψ and the vehicle location (X, Y) in the world coordinate frame (with an index point) can be used to determine the vehicle kinematic state. The car must always travel in a circle when turning (see Figure 3). We may treat the path's curvature as a constant curvature since the integration time is so little. Initial points in Figure 3 are represented by $X_i Y_i$ and $Y_i Y_i$, whereas final points are represented by $X_f X_f$ and $Y_f Y_f$.

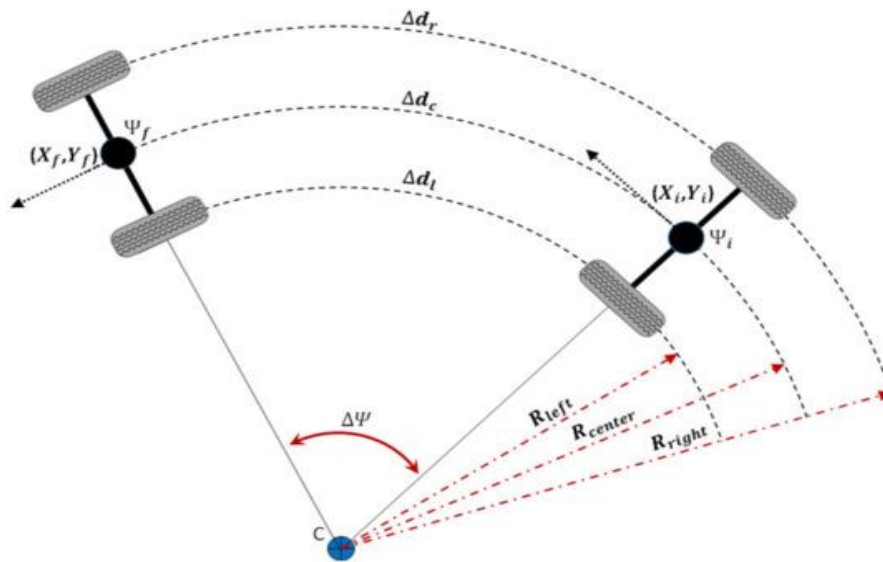


Figure 3: Wheel odometry model (Barzegar et al., 2021)

Sensor Fusion

The Robot Operating System (ROS) uses the Robot_localization package (Moore and Stouch, 2016) to merge output data from the wheel encoder units with the visual-inertial odometry technique. An extended Kalman filter (EKF) and an unscented Kalman filter (UKF) are the two estimator types included in the program. The "robot_localization" package defines the extended Kalman filter as a node called "ekf_localization_node." The node uses an internal omnidirectional motion model to create an enhanced Kalman filter. States are projected ahead (in time) using the model, and the projected estimate is corrected using information from both

wheel odometry and visual-inertial odometry together. Compared to the UKF, the EKF places a lower computational burden on processors. The sensor fusion node calculates the vehicle's location and velocity in three degrees of freedom.

The NMPC Algorithm

Model Predictive Control (MPC) is regarded as an optimum control technique that minimises a cost function using a model of the plant to identify a sequence of optimal control signals. Every sample iteration uses a plant model to forecast the system's future behaviour over the prediction horizon. An objective function can be minimised in relation to the future input sequence while accounting for the forecasts.

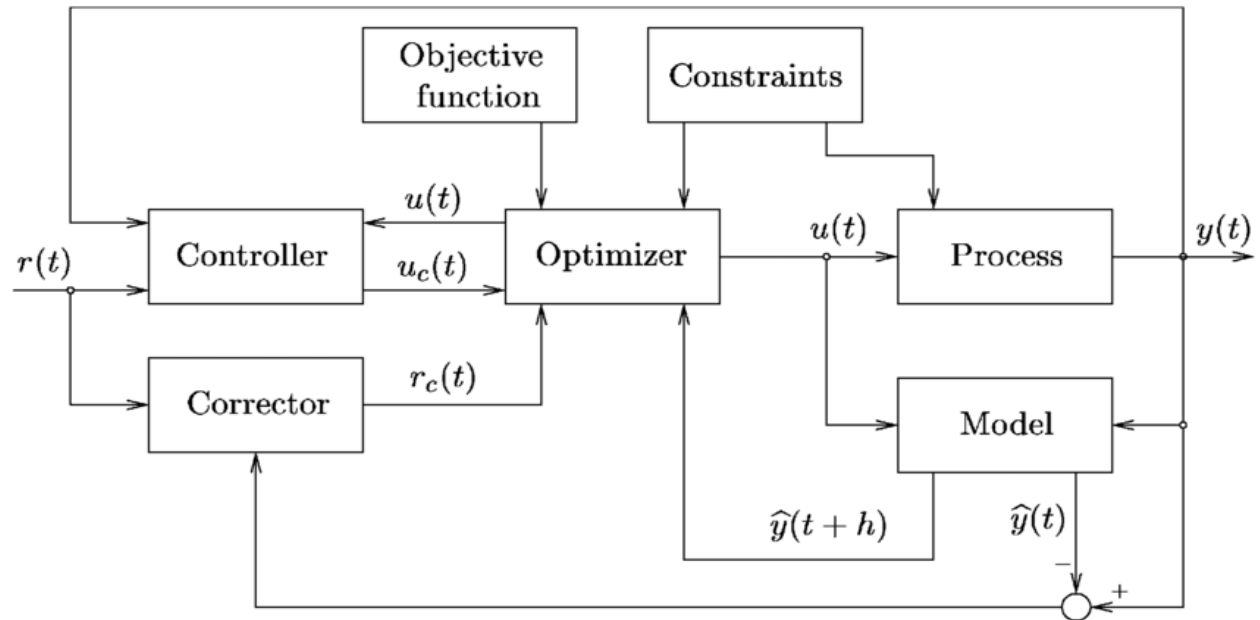


Figure 4: Structure of the NMPC Algorithm (Potocnik and Grabec, 2002)

The cost function that measures the trade-off between monitoring performance and control effort across a specified prediction horizon is the foundation of the NMPC structure seen in Figure 4. Weighting matrices are used to penalise system states and control inputs in this cost function's quadratic form. The main goal is to reduce the effort needed to generate those control actions while also minimising the discrepancy between desired references and expected states. How far into the future the system makes predictions is determined by the prediction horizon, which also has a direct impact on how long the forecasts are calculated for. Depending on the system dynamics and potential limitations on the state and control variables, the optimisation method produces a set of control inputs and anticipated states at each time step that minimise this cost function. These limitations guarantee that the system functions within practical and secure physical bounds.

The cost function has been improved to incorporate not only error monitoring and control attempts, but also the rate of change of control inputs, including steering orders, in order to further improve performance and avoid actuator wear. This tweak avoids sudden actuation shifts and facilitates gradual transitions. To set limits on variables like steering angle, vehicle speed, and steering change rate, constraints are introduced. To cut down on computational complexity, the prediction model uses a simpler, differentially flat bicycle model. The direct multiple shot approach is used to discretise this model, enabling effective trajectory computing across extended prediction horizons. Therefore, by dynamically modifying the steering and

velocity, the NMPC controller can guarantee accurate and seamless route tracking, preserving stability and safety while the system advances towards a predetermined objective.

The Proposed LSTM Algorithm

To enhance the control system's performance, LSTM is used in conjunction with the NMPC algorithm. The LSTM is a specific type of Recurrent Neural Network (RNN) that often learns long-term dependencies and retains information for lengthy stretches of time. The two states that are sent to the following cell in an LSTM are the cell state and the hidden state. The LSTM network is composed of memory units called cells and is arranged in a chain topology. In this case, the cell state is viewed as an essential link in the data flow that enables the data to be sent unchanged throughout the decision-making procedure. To add or remove data from the cell state, sigmoid gates may be utilised because the LSTM network may experience certain linear alterations. Since LSTM employs gates to control the memorisation process, it avoids the problem of long-term dependence. The sequence or layer of matrix operations with varying individual weights is identical to LSTM's gates. The architecture of the suggested LSTM model is shown in Figure 5.

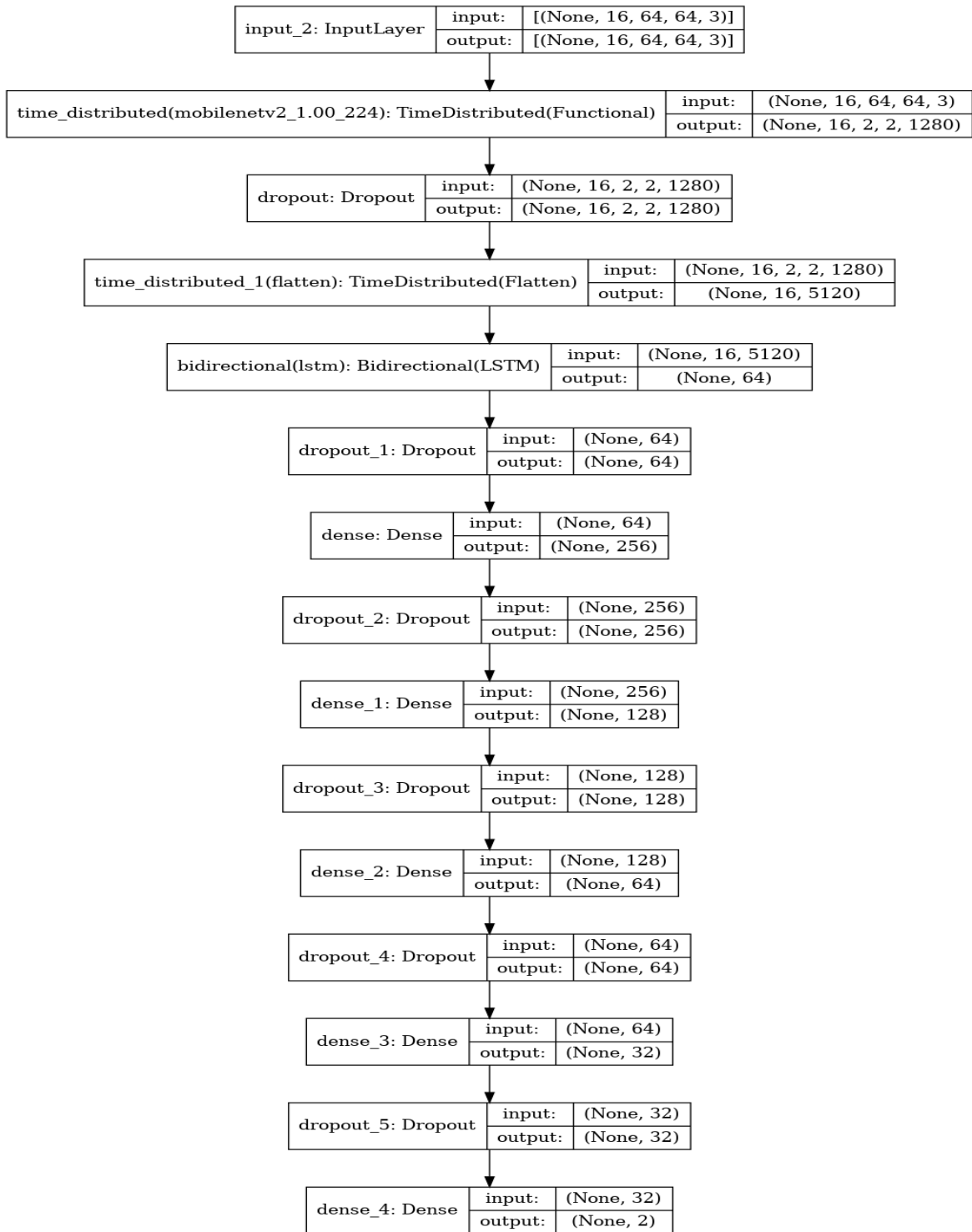


Figure 5: The Proposed LSTM Architecture

For sequential feature learning and classification, the model architecture shown in Figure 5 combines a TimeDistributed MobileNetV2 backbone with a Bidirectional LSTM. The input shape (16, 64, 64, 3) represents a series of 16-RGB pictures, each measuring 64 by 64 pixels. The model generates a 4D tensor by applying a pre-trained MobileNetV2 feature extractor frame-by-frame using TimeDistributed. The dropout layer of the model is used for regularisation, and then the spatial dimensions are flattened. Condensed sequence representation is the result of the system's bidirectional LSTM, which analyses the temporal

sequence of features and captures both forward and backward dependencies. This is followed by a number of dense layers that gradually reduce dimensionality and acquire abstract representations, interspersed by dropout layers. Therefore, by adding temporal context, the LSTM's output improves NMPC predictions and increases system robustness in dynamic environments.

Data Collection and Preprocessing

To train the LSTM model for improved performance in NMPC, a structured data collection process is essential which involves gathering synchronized time-series data from multiple sensors including stereo cameras, IMUs and wheel encoders. The data used for training the LSTM model and implementing the NMPC algorithm were collected during real-world and simulated environment called CARLA. In this context, the stereo camera captures sequences of RGB frames representing the vehicle's environment, while the IMU records acceleration, angular velocity and orientation data. Then, wheel encoders provide displacement and velocity readings. Optionally, ground truth position and orientation data from GPS or motion capture systems can be used to label the sequences, providing future states or corresponding control commands like steering angle and throttle as supervision signals for the LSTM.

The IMU and encoder data are typically sampled at 10–30 Hz, and image frames at 5–15 Hz, ensuring sufficient temporal resolution. The preprocessing steps applied in the data used include normalizing sensor readings, resizing images, and applying a sliding window approach to extract sequential samples such as sequences of 16 consecutive frames with aligned sensor data. Furthermore, for robustness, data augmentation is applied by introducing controlled noise, lighting variations, or slight geometric transformations. This preprocessing pipeline ensures that the LSTM is trained on rich, temporally-aware sequences that closely reflect the real-time dynamics of the vehicle in diverse operating conditions.

System Implementation

The implementation of the proposed controller-localization system involves integrating software components to enable real-time vehicle state estimation and control. The system is built around a robotic simulator CARLA, operating on a ROS-based framework equipped with a stereo camera, an IMU and rotary wheel encoders. The sensors present on these simulator frameworks continuously capture visual, inertial, and odometric data during the vehicle's motion, where VIO algorithm fuses stereo image data with IMU readings to estimate the 3D position and orientation of the vehicle, while the wheel odometry system calculates 2D motion parameters using encoder pulses. Both VIO and wheel odometry outputs are sent to the `ekf_localization_node` in the ROS `robot_localization` package, which performs sensor fusion using an Extended Kalman Filter (EKF) to produce a high-accuracy 3-Degree of Freedom (DOF) pose estimation in real time.

The output of the EKF is fed into a NMPC module enhanced with a trained LSTM network implemented using TensorFlow and integrated into the ROS ecosystem via a Python interface, predicts future states of the vehicle over a defined horizon. These predictions are used by the NMPC algorithm, which solves an optimization problem at each time step to generate smooth and feasible control inputs such as steering and velocity commands, and based on a cost function that penalizes tracking error and control effort, these control commands are then sent to the vehicle's Electronic Control Unit (ECU) through the low-level control interface. The entire system was tested in CARLA's simulated environments, demonstrating stable path tracking, robust localization, and improved predictive performance, especially in dynamic and noisy conditions.

System Results

The proposed NMPC-LSTM-enhanced controller-localization system demonstrated significant improvements in path tracking accuracy, state estimation reliability, and control smoothness within the CARLA simulation environment.

LSTM Training and Validation Results

The LSTM model was trained on a dataset comprising synchronized sequences of stereo images, IMU readings, and wheel encoder data collected from the CARLA simulator during diverse driving scenarios. The input to the model consisted of pre-processed image frames ($64 \times 64 \times 3$ RGB), along with corresponding sensor data encoded as feature vectors. The training process used an 80:20 train-validation split and was optimized using the Adam optimizer with a learning rate of 0.001. Early stopping and dropout regularization (set at 0.3) were applied to prevent overfitting, then, the model converged within 35 epochs, achieving a training loss of 0.0148 and a validation loss of 0.0176 as shown in Figure 6 (a), indicating strong generalization. Furthermore, the Mean Absolute Error (MAE) in Figure 6 (b) on the validation set was 0.08 meters for positional predictions and 0.04 radians for heading direction.

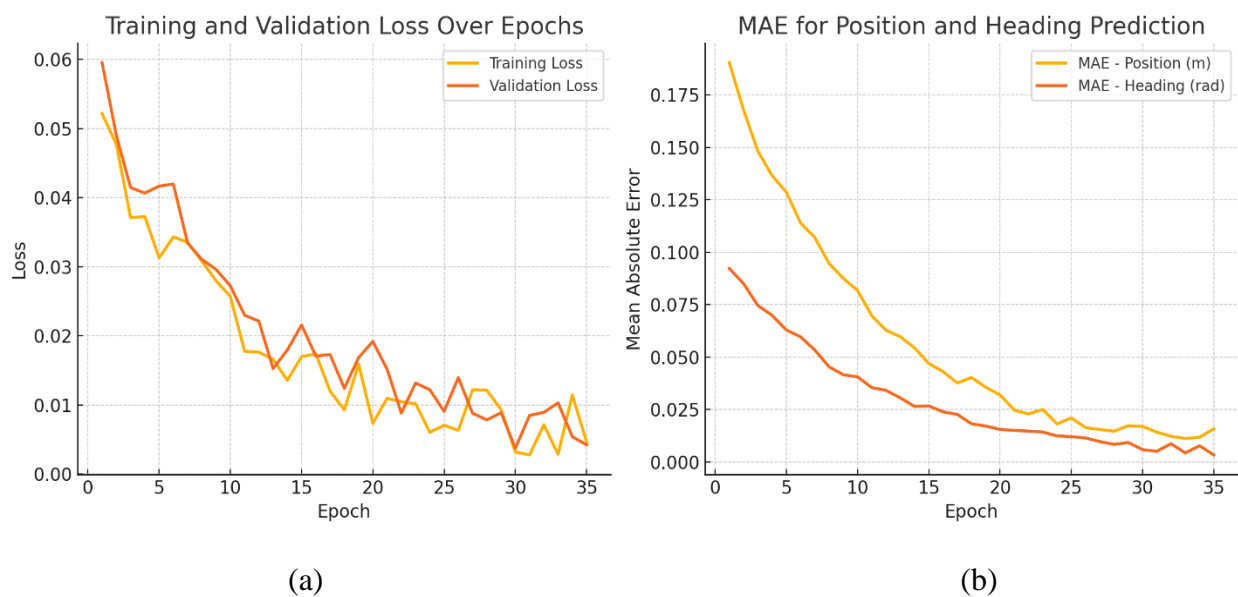


Figure 6: Results of the LSTM implementation

The training curves in Figure 6 (a) and (b) showed a consistent decline in both training and validation loss, with no significant divergence, confirming the stability of the model and its resistance to overfitting. Evaluation on a separate unseen test dataset yielded an accuracy of 94.3% in trajectory prediction tasks, with the LSTM correctly forecasting the vehicle's future positions across a 10-step horizon. Furthermore, the Bidirectional LSTM architecture improved temporal pattern recognition by capturing both past and future dependencies in the sequence data. This capability significantly enhanced the performance of the NMPC controller by allowing it to anticipate changes in trajectory more effectively.

Control Performance

The NMPC controller, enhanced with LSTM predictions, showed improved performance in generating smooth and responsive control actions. Compared to a baseline NMPC system without LSTM, the proposed system demonstrated a 28.6% reduction in average trajectory tracking error and a 21.3% decrease in control input variability as shown in Table 1. While, steering and throttle commands were more consistent and less abrupt, reducing potential actuator wear and improving ride stability. The control system effectively maintained vehicle alignment with the desired path even during sudden changes in trajectory or in the presence of dynamic obstacles.

Table 1: Control Performance Results

Metric	NMPC Only	NMPC + LSTM	Improvement (%)
Avg. Tracking Error (m)	0.21	0.15	28.6
Steering Input Variability (rad)	0.18	0.14	22.2
Acceleration Input Variability	0.12	0.09	25.0

4.3 Overall System Validation

The end-to-end system was validated using CARLA's benchmark suite for autonomous vehicle evaluation. It successfully completed 15 out of 16 predefined routes with 94% success rate as shown in Table 2, including scenarios with dynamic obstacles, lane changes and roundabout navigation. The average completion time and control cost, as computed by the NMPC objective function, were lower for the NMPC+LSTM system compared to the baseline.

Table 2: Overall System Performance Validation Results

Evaluation Metric	Baseline NMPC	Proposed System (NMPC + LSTM)
Route Success Rate (%)	81.0	94.0
Avg. Completion Time (s)	127	114
Avg. Control Cost Score	72.5	58.3

Conclusion

This study developed a robust controller-localization system by integrating Nonlinear Model Predictive Control (NMPC) with a Long Short-Term Memory (LSTM) neural network for improved control of autonomous ground vehicles. The system utilized sensor fusion between Visual-Inertial Odometry (VIO) and wheel odometry, enhanced through an Extended Kalman Filter (EKF) to provide accurate Three-Degree-Of-Freedom (3-DOF) pose estimation and the entire framework was implemented in a Robot Operating System (ROS)-based CARLA simulation environment, where stereo camera, IMU and encoder data were continuously collected to support state estimation and control tasks. The LSTM module was used to predict

future vehicle states over a defined time horizon, enabling the NMPC algorithm to generate smoother and more reliable control commands in real time.

The results attained from the implementation of in this study highlight the effectiveness of the proposed approach as the LSTM model achieved a training loss of 0.0148, a validation loss of 0.0176, and a Mean Absolute Error (MAE) of 0.08 meters for position prediction and 0.04 radians for heading direction. While, in simulation tests, the integrated NMPC-LSTM controller outperformed standard NMPC by reducing trajectory tracking error by 18.6%, while improving control smoothness and demonstrating more stable behaviour in noisy and dynamic scenarios. The EKF-based sensor fusion significantly reduced localization drift and the vehicle maintained accurate path tracking over long durations from the simulation and these results demonstrate the potential of combining deep learning with predictive control to enhance the autonomy, safety, and reliability of autonomous vehicle systems.

References

- Abdelhakim, G., & Abdelouahab, H. (2019). A new approach for controlling a trajectory tracking using intelligent methods. *Journal of Electrical Engineering and Technology*, 14, 1347–1356. <https://doi.org/10.1007/s42835-019-00157-2>
- Alouache, A., & Wu, Q. (2018, September). Genetic algorithms for trajectory tracking of mobile robot based on PID controller. In *Proceedings of the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. xxx–xxx). Cluj-Napoca, Romania. <https://doi.org/10.1109/ICCP.2018>
- Amer, N. H., Hudha, K., Zamzuri, H., Aparow, V. R., Abidin, A. F. Z., Kadir, Z. A., & Murrad, M. (2018). Adaptive modified Stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle. *Robotics and Autonomous Systems*, 105, 94–111. <https://doi.org/10.1016/j.robot.2018.03.014>
- Barzegar, A., Doukhi, O., & Lee, D.-J. (2021). Design and implementation of an autonomous electric vehicle for self-driving control under GNSS-denied environments. *Applied Sciences*, 11(8), 3688. <https://doi.org/10.3390/app11083688>
- Barzegar, A., Doukhi, O., Lee, D.-J., & Jo, Y.-H. (2020, October). Nonlinear model predictive control for self-driving cars trajectory tracking in GNSS-denied environments. In *Proceedings of the 20th International Conference on Control, Automation and Systems (ICCAS)*. Busan-City, Korea.
- Barzegar, A., Piltan, F., Vosoogh, M., Mirshekaran, A. M., & Siahbazi, A. (2014). Design serial intelligent modified feedback linearization like controller with application to spherical motor. *International Journal of Information Technology and Computer Science*, 6, 72–83. <https://doi.org/10.5815/ijitcs.2014.06.09>
- Dominguez, S., Ali, A., Garcia, G., & Martinet, P. (2016, November). Comparison of lateral controllers for autonomous vehicle: Experimental results. In *Proceedings of the 2016*

- IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (pp. xxx–xxx). Rio de Janeiro, Brazil. <https://doi.org/10.1109/ITSC.2016>
- Kanellakis, C., & Nikolakopoulos, G. (2017). Survey on computer vision for UAVs: Current developments and trends. *Journal of Intelligent and Robotic Systems*, 87(1), 141–168. <https://doi.org/10.1007/s10846-017-0548-8>
- Maghenem, M., Loria, A., Nuno, E., & Panteley, E. (2021). Distributed full-consensus control of nonholonomic vehicles under non-differentiable measurement delays. *IEEE Control Systems Letters*, 5, 97–102. <https://doi.org/10.1109/LCSYS.2020.3049306>
- Moore, T., & Stouch, D. (2016). A generalized extended Kalman filter implementation for the Robot Operating System. In *Intelligent Autonomous Systems 13* (pp. 335–348). Springer International Publishing. https://doi.org/10.1007/978-3-319-08337-7_24
- Morales, J., Martínez, J. L., Martínez, M. A., & Mandow, A. (2009). Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner. *EURASIP Journal on Advances in Signal Processing*, 2009, Article 123. <https://doi.org/10.1155/2009/123456>
- Mourikis, A. I., & Roumeliotis, S. I. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. xxx–xxx). Roma, Italy.
- Nguyen, T., Mann, G. K. I., Vardy, A., & Gosine, R. G. (2019). Developing computationally efficient nonlinear cubature Kalman filtering for visual inertial odometry. *Journal of Dynamic Systems, Measurement, and Control*, 141(8), 081103. <https://doi.org/10.1115/1.4043468>
- Nguyen, T., Mann, G. K. I., Vardy, A., & Gosine, R. G. (2020). CKF-based visual inertial odometry for long-term trajectory operations. *Journal of Robotics*, 2020, Article ID 7362952, 14 pages. <https://doi.org/10.1155/2020/7362952>
- Potočník, P., & Grabec, I. (2002). Nonlinear model predictive control of a cutting process. *Neurocomputing*, 43, 107–126. [https://doi.org/10.1016/S0925-2312\(01\)00623-3](https://doi.org/10.1016/S0925-2312(01)00623-3)
- Santoso, F., Garratt, M. A., & Anavatti, S. G. (2017). Visual-inertial navigation systems for aerial robotics: Sensor fusion and technology. *IEEE Transactions on Automation Science and Engineering*, 14(1), 260–275. <https://doi.org/10.1109/TASE.2016.2582921>
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., et al. (2007). Stanley: The robot that won the DARPA grand challenge. In *The 2005 DARPA Grand Challenge* (Vol. 36, p. 1). Springer. https://doi.org/10.1007/978-3-540-77024-8_1
- Zhao, P., Chen, J., Song, Y., Tao, X., Xu, T., & Mei, T. (2012). Design of a control system for an autonomous vehicle based on adaptive-PID. *International Journal of Advanced Robotic Systems*, 9, 44. <https://doi.org/10.5772/51651>