Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-20)

Real time obstacle motion prediction using neural network based extended Kalman filter for robot path planning

Najva Hassan[1], Abdul Saleem[2]
[1] *Dept. of Electrical and Electronics Engineering, Government Engineering College, Thrissur,*
[2] *Dept. of Electrical and Electronics Engineering, Government Engineering College, Thrissur,*
Corresponding author: abdulsaleempk@gmail.com

Abstract

Navigation for mobile robots in dynamic environments necessitates estimating the path of dynamic obstacles, which is accomplished in this study using an enhanced kalman filter. The measured data, however, contains bias and noise. The SDAE, a deep learning-based neural network structure, delivers noise-free data that the Kalman filter uses to construct an optimal measurement noise covariance matrix. This matrix is then used by the Kalman filter to estimate an error-free obstacle path.The SDAE is trained using both the Adam and stochastic gradient learning algorithms. To ensure safe navigation, the robot's path is re-planned based on the estimated obstacle path. Numerical simulations using MATLAB demonstrate that the novel methodology is more relevant and superior to traditional Kalman and Particle filter approaches, and that it can be applied in a variety of navigational applications. In terms of computing time and robustness in closely spaced obstacles, simulation testing indicated that path planning using the proposed technique excels the hybrid A star, artificial potential field, and decision algorithms.

Keywords: Denoising autoencoder; dynamic path planning; Kalman filter; measurement noise covariance; motion prediction;

1. Introduction

As a result of recent robotics advancements, autonomous mobile robots are increasingly being employed in a wide range of applications, including military, hospitals, farm imaging, and surveillance. Mobile robots could operate in hazardous and unpredictably changing situations. Because the barriers are immovable in a static environment, path planning is rather simple, and offline path planning suffices. Path planning is a difficult problem in a dynamic environment with moving obstacles because the robot must re-plan its path to reach the destination without colliding.

To achieve intelligent navigation of mobile robots, sensor-actuator control methods are adopted. Most navigation approaches, including global navigation satellite systems and inertial navigation systems, use the Kalman filter (Wang S.L., 2013). A unique deep learning-based prediction method is developed in (Park, J.S., 2020) for generating collision-free trajectories for a robot working in an obscured environment near a human obstacle. In (Park, J.S., 2020), an occlusion-aware planner is employed to compute collision-free trajectories, resulting in improved human motion prediction accuracy. The Extended Kalman Filter and RGBD-SLAM are employed in order to solve landmark localization and build 2D and 3D maps of the environment (Khan, M.S.A., *et al.*, 2021). SLAM techniques are used on a two-wheeled mobile robot with an encoder to monitor feedback, and the robot is intelligently built to move autonomously in an indoor static environment. The authors of (Van Den Berg, *et al.*, 2005) employed road-maps for robot motion planning in dynamic scenarios. In this context, the local path planning has been developed using a depth-first search on an implicit grid. This method is applicable to any robot type in any configuration space, and the obstacle motion is unrestricted. Dynamic road maps, on the other hand, demand additional processes for smoothing the path prior to execution, making path re-planning difficult. To handle the path planning problem and to have continuous re-planning of the path, (Volz A., *et al.*, 2019) presents a predictive route following controller. The ideal control actions for traveling along the intended path are computed here, and the path is regularly re-planned. Another approach is the one proposed in (Lin X., *et al.*, 2020) , which incorporates artificial potential field and decision tree concepts. The improved artificial potential field method addresses the problem of local minima and thus enables real-time path planning. However, the robot experienced vibrations under the influence of closely spaced obstacles. To avoid high speed obstacles, a viable two period velocity obstacle algorithm is proposed in (Liu Z., *et al.*, 2018). The first period predicts potential collisions within a limited time horizon, while the second period predicts collisions beyond that horizon. The robot's dynamic model and moving impediments have not been taken into account, resulting in lower prediction accuracy. The hybrid simulated annealing approach is utilized in (Saricicek I., *et al.*, 2022) to determine autonomous vehicle routes. An energy efficient routing and scheduling system is also offered in (Saricicek I., *et al.*, 2022) to reduce the total energy spent by the cars by taking both the traveled distance and the vehicle's weight into account. By merging vision-based estimation and control loops, in (Roggeman H., *et al.*, 2017) safe and autonomous navigation of mobile robots is achieved. To estimate the position of moving obstacles, a method based on stereo vision data is used. For powerful computation, GPU assistance is needed. In (Lin Y., *et al.* , 2017), a sampling-based path planning approach is designed for the safe operation of an unmanned aerial vehicle. The planning time is reduced using a simplified node connection. In (Zhu, Q., *et.al*, 2019), a path planner based on a recurrent fuzzy neural network (RFNN) is created to plan the trajectory and motion of mobile robots in order to accomplish a target. To improve nonlinear programming performance, RFNN integrates fuzzy logic inference and neural network learning characteristics. To improve the autonomy and intelligence of autonomous guided vehicles (AGVs) navigation control, (Ren, Z., *et.al*, 2021) presented a hybrid real-time optimum control strategy based on deep neural networks (DNNs). The motion planning problem of an AGV with static and dynamic obstacles is presented as a nonlinear optimum control problem (OCP) in (Ren, Z., *et.al*, 2021), and the optimal solution is obtained using a direct method incorporating a smooth transformation methodology. The Prognostics-aware Multi-Robot Route Planning (P-MRRP) algorithm is proposed in (Yayan, U., *et.al*, 2021) for improving the robot team's lifetime. In the P-MRRP algorithm, routes are first created using a route set generation algorithm, and then the most reliable route set is chosen by calculating PoRC based on the robot team's reliability, as well as the effect of load on the robots' path.

In (Elnagar A., 2001), the Kalman filter is utilized to forecast the future positions and orientations of moving obstacles in dynamic situations. Under the assumption that the prior position and orientation are known, the Kalman filter may efficiently anticipate obstacle positions. (Wei, H., *et al.*, 2021) proposed a method for estimating motion state based on region-level instance segmentation and the extended Kalman filter (EKF). To create optimum motion parameters, the EKF model takes into account ego-motion and integrates it along with optical flow and disparity. The Kalman filter's prediction, on the other hand, is dependent on the process noise covariance matrix R and the measurement noise covariance matrix Q. When the measurement noise covariance matrix is chosen arbitrarily, the filtering accuracy degrades. In (Mehra R, 1971), an iterative approach for obtaining unbiased and reliable estimations of Q and R has been developed. However, this iterative method can be used only for the case in which the form of Q is known and the number of unknown elements in Q is less than $n \times r$, where n is the dimension of the state vector and r is the dimension of the measurement vector. The measurement noise covariance is identified in (Diversi R. *et al.*, 2005) without any knowledge of the noise mean by considering linear discrete stochastic systems. An estimation of the measurement noise covariance is done in (Yuen K.V., *et al.*, 2013) using a probabilistic method. In (Yuen K.V., *et al.*, 2013), the Bayesian technique has been utilized to determine the optimal noise parameter estimation and associated estimation uncertainty. The noise covariance of a scalar system is estimated using the maximum likelihood approach by the authors of (Matisko P., *et al.*, 2010). However, in (Matisko P., *et al.*, 2010), they implemented a simple searching strategy that would be prohibitively expensive for larger systems. In (Shumway R.H., *et al.*, 2019), the measurement noise covariance matrix is computed using a gradient-based numerical optimization ap-

proach that can be applied to measurements taken at irregular intervals but demands a lot of computing power. The authors of (Valappil J., *et al.*, 2000) have developed a method for estimating the noise covariance matrix of an extended Kalman filter based on Monte Carlo simulations. Using a priori knowledge of the uncertainties, samples of the parameters are generated in (Valappil J., *et al.*, 2000) and provided a simplified approach for tuning the Kalman filter. An auto covariance least square method is proposed by the authors of (Odelson B.J., *et al.*, 2006) to estimate the Q and R of Kalman filter. A lagged auto covariance function between the measurements is defined in (Odelson B.J., *et al.*, 2006), which is used to develop a linear least squares formulation to estimate Q and R. A wavelet transform is proposed in (Park S., *et al.*, 2019) to estimate the time-varying measurement noise variance. The noise covariance matrix can be correctly predicted using the wavelet transform approach. The computation time, on the other hand, is longer. The authors of (Wu F., *et al.*, 2020) use temporal convolutional neural networks to accurately evaluate the measurement noise covariance matrix. The sensor data sequences are used to estimate the noise covariance via neural networks. Changes in the environment can be reflected using temporal convolutional neural networks. The approach proposed in (Wu F., *et al.*, 2020), on the other hand, has a high training cost and cannot be learned directly on the resource constrained integrated navigation platform. The enhanced Hough Transform (HT) algorithm and the Least Squares (LS) method are combined in (Gao, *et.al*, 2018) as an effective methodology for multi-objective recognition in 8-ball billiards vision system. In (Ariff, M.A.M.,, 2021), a time-series prediction technique based on the nonlinear auto-regressive exogenous neural network (NARX) algorithm is developed to forecast generator speed deviations after a system disturbance. Using the developed strategy, the author of (Ariff, M.A.M.,, 2021) is able to speed up the overall coherency detection procedure in a power system operation.
According to the literature review, the majority of dynamic path planning algorithms assume that the obstacle motion is known in advance (Xidias, 2021) or that it moves at a constant velocity (Lin X., *et al.*, 2020). In the vast majority of circumstances, however, assuming obstacle motion is impossible. Most path planning algorithms require more time to re-plan (Xidias, 2021), resulting in higher processing time (Lin Y., *et al.* , 2017) and a significant amount of computing labor (Roggeman H., *et al.*, 2017). The literature review also reveals that, dynamic path planning algorithms that use sensors for motion prediction may fail to generate a precise collision-free path due to erroneous obstacle path predictions caused by noisy data. In this study, we offer an approach for estimating the motion of obstacles in dynamic conditions, which aids the robot in avoiding obstacles, is applicable to varying velocity, and requires less computing time with higher prediction accuracy. The Kalman filter is an excellent option for predicting obstacle paths. For accurate prediction, however, knowledge of the noise error covariance matrices is essential. Furthermore, on-line processing of these matrices is often necessary for any time-varying nonlinear system, such as a mobile robot. In contrast to the use of approximation or random selection, this method employs the SDAE to determine measurement noise covariance. The following are the significant contributions of this work:

- This research develops an approach for determining obstacle motion in dynamic environments using multi-layer neural networks that is suitable to varying velocity and takes less computing time with improved prediction accuracy. The deep learning based neural network structure proposed in this work is highly reliable and robust against noise.

- Once trained, the developed stacked denoising autoencoder based extended Kalman filter is able to predict the obstacle state in the presence of both Gaussian and non- Gaussian noise.

- In terms of performance metrics such as integral squared error (ISE), mean absolute error (MAE), and integral absolute error (ISE), the developed SDAE methodology with Adam optimizer outperforms the conventional Kalman filter, Particle filter, and denoising autoencoder (DAE)based Kalman filter for both colored and Gaussian noise. As compared to (Sedighi S., *et al.*, 2019), (Ge S.S., *et al.*, 2002), and (Xidias, 2021), the developed methodology generates an optimal path in terms of processing time, path length, and obstacle avoidance.

The rest of this paper is organized as follows: Problem formulation is explained in section 2. Section

3 describes the proposed algorithm for motion prediction. Simulation results are given in section 4. Finally, section 5 presents the concluding remarks.

2. Problem formulation

The path planning problem is defined as finding a collision free path for an autonomous vehicle from a given start position to a goal position, satisfying a set of constraints. Assume that the mobile robot moves in a two dimensional (2D) space. The objective of robot the path planning is to find a path from a start

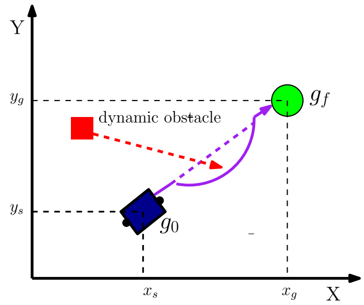


Fig. 1. Problem definition

position g_0 to a goal position g_f such that the robot avoids collision with obstacles. Let g represents the path which can be defined as

$$g = [g_0, g_1, g_2 \cdots g_{n-2}, g_{n-1}, g_f] \tag{1}$$

where $g_1, g_2 \ldots g_{n-1}$ are the via points.
To ensure that the path is collision free, there should be no static and dynamic obstacle in the robot's safety zone at any time i.e.,

$$\text{for } i = 1, 2, \ldots, N, \ o_{pi}(t) \notin P(x(t)) \tag{2}$$

where N is the number of obstacles, $P(x(t))$ corresponds to the safety zone of the robot and o_{p_i} is the position of the obstacle. The state of the j^{th} obstacle is given by

$$o_j(t) = \begin{bmatrix} o_{p_j}(t) \\ o_{v_j}(t) \end{bmatrix} \tag{3}$$

where $o_{v_j}(t)$ is the velocity of the obstacle. Considering an obstacle with constant velocity, the relation between the position and velocity of the j^{th}obstacle using basic kinetic formula can be expressed as (Lin Y., *et al.* , 2017)

$$o_{p_j}(t) = o_{p_j}(t_0) + o_{v_j}(t_0) * (t - t_0) \tag{4}$$

The state space model of a robot can be represented as

$$\dot{r}(t) = f\,(r(t), u(t)) \tag{5}$$

where $r(t)$ is the state of the robot and $u(t)$ corresponds to the control vector. Besides the condition of collision free, the path should be shortest also which can be expressed mathematically as

$$g^\star = \text{argmin} \int_g dq \tag{6}$$

where dq is the differential of arc length of the path. In short, the problem can be defined as: Find a continuous path $g(x, y)$ from the start position $g_0(x_s, y_s)$ to the goal position $g_f(x_g, y_g)$ satisfying the constraints given by Equations (2), (4), and (5). These concepts are shown in Fig. 1

3. Proposed methodology

In a real world scenario, the robots are supposed to navigate in dynamic environments which consist of both static and dynamic obstacles. Obstacle motion prediction is a critical issue in dynamic path planning. While addressing the motion planning problem, uncertainty in the obstacle motion needs to be considered. The knowledge about obstacle motion information is very essential for the robots to complete their task effectively and safely. In most of the robot path planning algorithms, it is assumed that the obstacles move with a constant velocity or their positions are known to the robots. However, the data obtained using the sensors may not be precise and can be noisy. Hence, the goal of a successful robot navigation can be affected. The commonly adopted approach in navigation system for the obstacle path prediction is the use of extended Kalman filter. The prediction accuracy of the Kalman filter is greatly affected by the choice of measurement noise covariance matrix R. Filtering techniques and shallow neural networks such as denoising autoencoder (DAE) (Park S., *et al.*, 2019) for removing the noise have limited performance in the presence of noises other than Gaussian. In this work, a SDAE is proposed to obtain an optimum measurement covariance matrix which is used in an extended Kalman filter to estimate the states of the moving obstacle accurately. Adam and stochastic gradient descent (SGD) algorithm are used as the training algorithm to achieve maximum accuracy with reduced computation time.

3.1 Stacked Denoising Autoencoder Based Extended Kalman Filter

Kalman filter is a powerful tool for the state estimation of a system. It can provide a more accurate estimate even if the measurements are noisy. Kalman filter is capable of online real time processing and hence it can be used to estimate the position and velocity of moving obstacles in path planning problems. Kalman filter operates in two steps

- Prediction - Based on the past sensor data the next values are predicted.
- Updation - To obtain a value closer to the actual value, the predicted value is refined using the measured value.

The Kalman filter works well for the linear functions. However, obstacle motion paths can be nonlinear and so this work considers an extended Kalman filter for the obstacle path estimation. In the extended Kalman filter, the nonlinear equation is linearised using Jacobian matrix (Prevost C.G., *et al.*, 2007). Consider a moving robot car having the state

$$r_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \tag{7}$$

where x_k, y_k, and θ_k corresponds to the x position, y position, and the orientation of the moving robot car respectively. The state space model of a robot car after linearisation is given by

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = A \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + B \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix} + v_{k-1} \tag{8}$$

where $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} \cos\theta_{k-1} * dk & 0 \\ \sin\theta_{k-1} * dk & 0 \\ 0 & dk \end{bmatrix}$, and $v_{k-1} = \begin{bmatrix} noise_{k-1} \\ noise_{k-1} \\ noise_{k-1} \end{bmatrix}$

The state at time step k is computed using the state space model, state estimate, and the control input vector at the previous time step (k-1)

$$\hat{r}_k = f(r_{k-1}, u_{k-1}) \tag{9}$$

The observation model is defined as

$$z_k = Hr_k + w_k \tag{10}$$

where w_k is the sensor noise and H matrix has the same number of rows as sensor measurements and the same number of columns as states. In a robot car model, the H matrix is defined as

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The updated state $\hat{r_k}'$ is calculated from

$$\hat{r}'_k = \hat{r}_k + K(z_k - H_k\hat{r_k}) \tag{11}$$

where K is the Kalman gain which is obtained using

$$K = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \tag{12}$$

where R_k is the covariance of the sensor noise. Here P_k is the error covariance matrix and it is first predicted using

$$P_k = F_k P_{k-1} F_k^T + Q_k \tag{13}$$

where Q_k is the process noise covariance, F_k is equivalent to the A matrix in Equation (8) and then updated with

$$P'_k = P_k - KH_k P_k \tag{14}$$

From the above equations, it is clear that sensor noise covariance R and process noise covariance Q are important factors that determine the extended Kalman filter performance. For most of the cases, R is assumed to be constant or adjusted manually by trial and error approach. However, this may affect the performance of the extended Kalman filter and can result in an inaccurate estimation of the obstacle motion. A multi layer neural network based method is developed to estimate the obstacle state accurately. SDAE are used to denoise the sensor data. The measurement noise covariance matrix is calculated from the measured data and the noise free data obtained using the SDAE. The adaptively determined measurement noise covariance matrix is further used by the extended Kalman filter for predicting the obstacle state accurately. The training of the SDAEs is given in Algorithm 1 and the multi layer neural network based algorithm for estimating the measurement noise covariance R is described in Algorithm 2. The learning based estimation of noise covariance matrix R consists of three steps.

1. Train the neural network using a set of input-output data. A set of noise free data, S_{mi}, i=1, 2, 3, ..., n where n is the length of training data is collected which are considered as the target data of the neural network. Let T_i , be the data obtained by adding noises to S_{mi}. Both Gaussian noise and colored noise are considered in this work. Then T_i represents the input data to the neural network. The length of training data n is so chosen that the cost function C finally converges to zero. The trained DAE are stacked together such that maximum accuracy is achieved.

2. Apply the noisy measured real time data to the trained SDAE. Then the output of the neural network will be a noise free data S_{nf}.

3. Compute the measurement noise covariance matrix R using

$$R = \begin{bmatrix} \Delta x^2 & 0 & 0 \\ 0 & \Delta y^2 & \\ 0 & 0 & \Delta v^2 \end{bmatrix} \tag{15}$$

Where Δx is the difference between measured x-position and noise free x-position, Δy is defined as the difference between measured y position and noise free y position, and Δv is defined as the difference between measured velocity and noise free velocity.

Algorithm 1: Training of the SDAEs

Training;
Require
Target: Noise free data S_{mi}, $i = 1, 2, 3, \ldots, n$, n is the length of training data;
Input: Noise is added to the noise free data S_{mi} to obtain the input data;
α: Step size;
β_1, β_2 : Exponential decay rates for the moment estimates;
$C(\theta)$: Stochastic objective function with parameters θ;
θ_0: Initial parameter vector;
m_0: Initialize first moment vector;
v_0: Initialize second moment vector;
t: Initialize time step;
while θ_t *not converged* **do**
  $t \leftarrow t + 1$;
  $g_t \leftarrow \Delta_\theta f_t(\theta_{t-1})$ (Get gradients objective at timestep t);
  $m_t \leftarrow \beta_1.m_{t-1} + (1 - \beta_1).g_t$ (Update biased first moment estimate)
  $v_t \leftarrow \beta_2.v_{t-1} + (1 - \beta_2).g_t^2$ (Update biased second raw moment estimate);
  $\hat{m}_t \leftarrow \frac{m_t}{(1-\beta^t)} g_t^2$ (Compute bias-corrected first moment estimate);
  $\hat{v}_t \leftarrow \frac{v_t}{(1-\beta_t^2)}$(Compute bias-corrected second raw moment estimate);
  $\theta_t \leftarrow \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$ (Update parameters);
  end while;
  return θ_t (Resulting parameters)
end

Algorithm 2: Online estimation of measurement noise covariance matrix R

Begin
Step 1: **Input**: Sensor data S_n
Step 2: Give the input to the trained SDAE "net$_\theta$"
for$(t = 0 : t_s)$
Step 3: Obtain the output

$$S_{nf} = net_\theta(S_n)$$

Step 4: Obtain

$$\Delta x = S_{nf}(x) - S_n(x)$$

$$\Delta y = S_{nf}(y) - S_n(y)$$

$$\Delta v = S_{nf}(v) - S_n(v)$$

Step 5: Calculate the measurement noise covariance using

$$R = \begin{bmatrix} \Delta x^2 & 0 & 0 \\ 0 & \Delta y^2 & 0 \\ 0 & 0 & \Delta v^2 \end{bmatrix}$$

Step 6: Return R
end

3.1.1 Stacked Denoising Autoencoders

Denoising autoencoders are neural networks which are the extension of autoencoders (Xing C., *et al.*, 2016). They are trained to obtain the original data from the corrupted version of it. A DAE consists of encoder-decoder and a set of hidden layers similar to that of a conventional autoencoder. But the input to the DAE is corrupted data and the decoder output is the noise free data. The working of the DAE is shown in Fig. 2. For training, a set of noise free measured data is obtained. Then the input signal $\hat{a}$ is

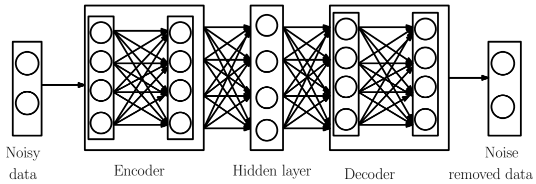


Fig. 2. Denoising autoencoder

obtained by adding noise to the noise free data, a. The noisy data $\hat{a}$ is mapped through the encoder to the hidden layer. The output of the neurons in the hidden layer is given by

$$h = f_e(W_{ih}\hat{a} + b_{ih}) \tag{16}$$

W_{ih} is the weight matrix connecting the input layer and hidden layer, f_e is the activation function of encoding layer, and b_{ih} is the bias in the hidden layer. The original data is reconstructed by the decoder through the hidden layer.

$$a_e = f_d(W_{ho}h + b_{ho}) \tag{17}$$

W_{ho} is the weight matrix connecting the output layer and hidden layer, f_d is the activation function of decoding layer, and b_{ho} is the bias in the output layer. The reconstruction error in a DAE is calculated as

$$C(a, a_e) = ||a - a_e||^2 \tag{18}$$

where a_e is the output. The cost function is minimized with respect to the DAE model weights

$$\theta = arg_\theta \; min \frac{1}{n}\sum_{i=1}^{n} C(a^{(i)}, a_e^{(i)}) \tag{19}$$

where θ corresponds to (W, b) and C is the cost function.

The DAEs are robust and provides better results when trained properly. However, its capabilities are limited and often do not perform well for data with large noise. Thus a SDAE is used in this paper. SDAEs are built by stacking DAE and have more than one hidden layer (Vincent P., *et al.*, 2010). It consists of two encoding layers and two decoding layers. The output of the first encoding layer is given as the input data to the second encoding layer. In this work, a data set of 5000 samples are used to train the SDAE. The additive white gaussian noise and the colored noise are added to the data set which gives the input data for training purpose. The developed SDAE consists of two hidden layers with 20 neurons in each layer. Initially, the first DAE is trained and the weights w, bias b and features h are obtained. These features h are provided as the input to the next encoding layer. Layer wise training of DAE is performed and are stacked together. Adam and stochastic gradient descent algorithms are used as the optimization algorithms for learning. The gradient estimate is computed by using a loss function in the stochastic gradient descent algorithm. The learning rate determines the magnitude of the parameter updation. Choosing of the learning rate is a non trivial task in stochastic descent algorithm. The advantages of both adaptive gradient and RMSprop algorithms are combined in an Adam optimizer. The adam algorithm updates the gradient (m_t) and squared gradient (v_t) , with the hyper-parameters β_1, β_2controlling the exponential decay rates of these moving averages. The moving averages are estimates of the gradient's first moment (the mean) and second raw moment (Soydaner, D.,, 2020). The pseudo code ofthe Adam algorithm is explained in Algorithm 1. It works efficiently for problems with noisy and sparse gradients. The SDAE based extended Kalman filter is used to estimate the path of moving obstacle, which is explained in Algorithm 3.

Algorithm 3: Proposed SDAE based extended Kalman filter for obstacle motion prediction

Begin

Step 1: Input trained SDAE net_θ, noisy data S_n.

Step 2: Obtain noise free data S_{nf}.

$$S_{nf} = net_\theta(S_n)$$

Step 3: Calculate the measurement noise covariance R .

$$R = \begin{bmatrix} \Delta x^2 & 0 & 0 \\ 0 & \Delta y^2 & 0 \\ 0 & 0 & \Delta v^2 \end{bmatrix}$$

Step 4: Adjust the Kalman gain K.

$$K = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

using updated R.

Step 5: Estimate the moving obstacle state

$$\hat{r}'_k = \hat{r}_k + K(z_k - H_k \hat{r_k})$$

end

3.2 Path planning in dynamic environments

In real time applications, the environment that a robot has to navigate can be static or dynamic. If the environment is dynamic then the robot should be able to predict the obstacle motion so as to successfully avoid a possible collision with the obstacle. The schematic diagram of the proposed method for path planning in a dynamic environment is shown in Fig. 3. The developed method is divided into two phases. Initially, the path is planned considering the static obstacles. In the second phase, the obstacle motion is predicted and the robot path is re-planned so that the collision is avoided.

3.2.1 Initial path generation

Initially, an offline path planning is done assuming that the environment is static. Let the start and goal position be g_0 and g_f respectively. In this approach, we are assuming that the current position of the moving obstacles is known to us. Let the configuration space be C_{space}. It consists of a collision free space C_{fs} and a space with obstacles C_{obs}. Randomly choose a set of configurations P and check collision at each selected n closest neighbor points. Thus the shortest path is calculated initially using the algorithm proposed in (Chen J., *et al.*, 2019) within a time period t.

3.2.2 Obstacle motion prediction and path re-planning

In this work, Algorithm 3 is used to predict the obstacle motion. The obstacle path is predicted for the given time horizon t which is the time required to calculate the initial path. Now check if an intersection of the initially planned robot path and the estimated obstacle path exists or not. If an intersection of the two paths occurs then the robot path is re-planned. The new path is now the current robot path and the process of checking obstacle path and robot path is continued and re-planning is done when both paths intersect until the goal position is reached.

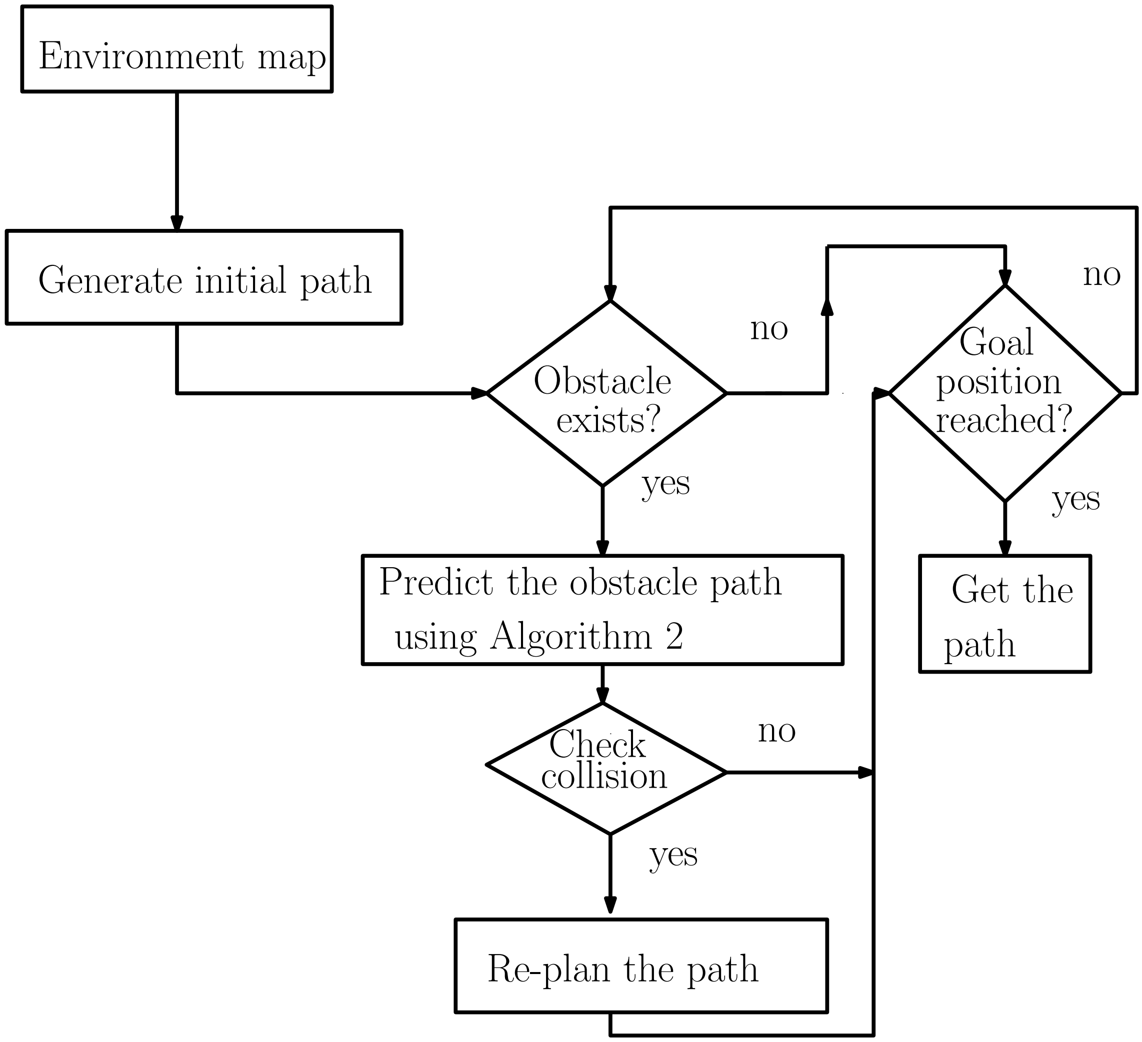


Fig. 3. Schematic diagram of proposed method for path planning in dynamic environments

4. Results and discussions

In this section the effectiveness of the developed algorithm for predicting the obstacle motion is validated using various simulations. A comparative assessment of the prediction algorithm is also performed by comparing with conventional Kalman filter, Particle filter and denoising autoencoder based Kalman filter. In order to assess the efficacy of the proposed method, various performance metrics such as IAE, ISE and MAE in the obstacle path prediction are analyzed.

$$ISE = \int_0^t e(t)^2 dt \tag{20}$$

The accumulated error is denoted by the integral of absolute error and is obtained by

$$IAE = \int_0^t |e(t)| dt \tag{21}$$

where e(t) is the difference between the obstacle's actual and estimated path. The performance of the algorithm is tested and validated for both static and dynamic obstacles. The performance of the proposed algorithm is evaluated using MATLAB simulated environments by comparing it with path planning algorithms (Sedighi S., *et al.*, 2019),(Ge S.S., *et al.*, 2002), and (Xidias, 2021).

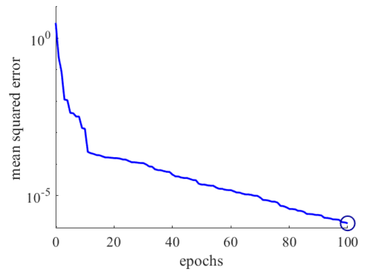


Fig. 4. Performance plot of of neural network

4.1 Neural Network training

The objective of neural network training is to generate SDAEs which gives a noise free data from a noisy data. MATLAB 2020a is used in this work to implement the SDAE. The pioneer-1 mobile robot data set is used for training the neural network. This noise free data set consists of sensor readings of pioneer-1 mobile robot which are the targets or desired outputs of neural network. The input to the neural network during the training is obtained by adding noises to the pioneer 1 data. We have considered both colored and white noises. The deep neural network structure used here consists of two hidden layers. The weights and bias are tuned using both Adam and stochastic gradient descent algorithms. The sigmoid function is used as the activation function for all the layers. Once the neural network is trained, the SDAEs will provide a noise free data if a noisy data is given as input to it. The parameters for training the SDAEs are given in Table 1. The performance plot which is the variation of the training record error values against the number of training epochs is shown in Fig. 4. At the end of the training phase, mean squared error reaches a value of order 10^-5. The small value of the mean squared error implies that the desired outputs and the neural networks outputs for the training set have become very close to each other.

Table 1. Parameters for training stacked denoising autoencoder

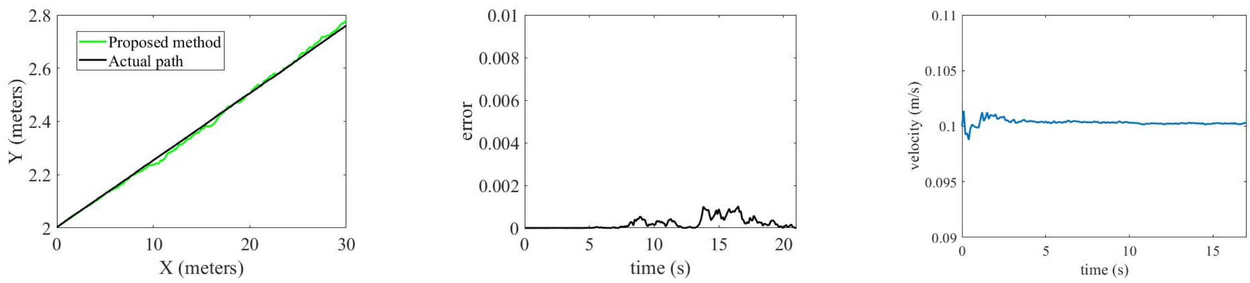
Parameters	**Value**
Learning rate	0.02
Number of epochs	100
Number of training data sequences in each iteration	100
Learning algorithm	Adam

The trained SDAEs are used to find the measurement noise covariance of the extended Kalman filter for estimating the obstacle path. The proposed algorithm is implemented on i7 core, 32gb laptop. The performance of the proposed SDAE based extended Kalman filter for estimating the obstacle path is discussed subsequently.

4.2 Performance of the stacked denoising autoencoder based extended Kalman filter

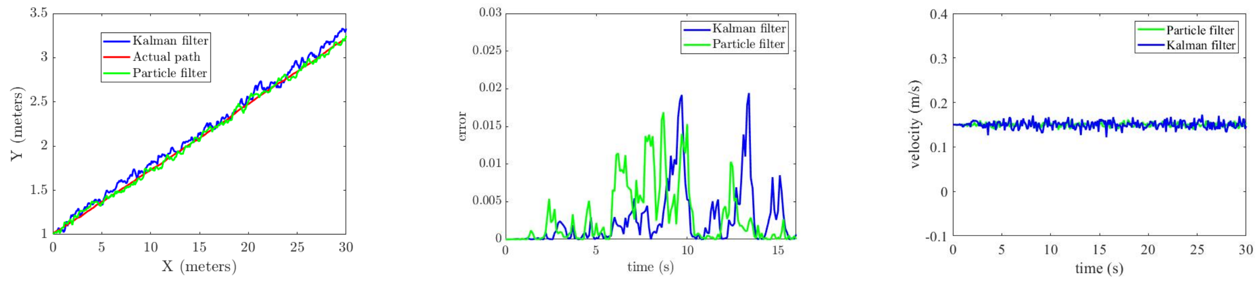
In this work, the role of the extended Kalman filter is to estimate the obstacle path. The accuracy of prediction using extended Kalman filter is dependent on the Kalman gain which further depends on the measurement noise covariance matrix. The SDAEs are trained using Algorithm 1 and are used to estimate the measurement noise covariance matrix using Algorithm 2 described in section 3. Initially, an obstacle moving with a constant velocity is considered.

The initial position of the moving obstacle is measured and is given as input to the trained SDAEs. Then the output of SDAEs gives noise free measured data. Now the measurement noise covariance



(a) Obstacle path **(b)** Estimation error **(c)** Velocity estimation

Fig. 5. Performance of the SDAE based Kalman filter



(a) Obstacle path **(b)** Estimation error **(c)** Velocity estimation

Fig. 6. Performance of the conventional Kalman filter and Particle filter

matrix can be found using Equation (15), which is computed as $R = \begin{bmatrix} 0.012 & 0 & 0 \\ 0 & 0.015 & 0 \\ 0 & 0 & 0.023 \end{bmatrix}$
The Kalman gain is calculated by substituting the estimated measurement covariance matrix in Equation (12). The obstacle path is estimated using Equations (9)-(14) repeatedly. The estimated obstacle path is shown in Fig. 5a. The actual path of the obstacle is calculated theoretically by using the basic kinetic formula given by Equation (4) and it is plotted in the same figure. From 5a, it is clear that the estimated obstacle path using the proposed algorithm follows the actual path of the obstacle. The error in the estimated path which is computed as

$$\text{error} = \sqrt{(\text{actual path} - \text{estimated path})^2}$$

is plotted in Fig. 5b. The maximum error in estimation is of the order of $10^{-}3$ which is negligible and converges to zero. The velocity profile of the moving obstacle estimated using the SDAE based extended Kalman filter is shown in Fig. 5c. The estimated velocity of the moving obstacle remains constant with time and follows the actual velocity.

To evaluate the performance of proposed method, it is compared with conventional Kalman filter and Particle filter (Berntorp K., *et al.*, 2016). Fig. 6a shows the actual and estimated paths of an obstacle. It is obvious from this figure that the estimated path deviates from the actual path for both Kalman and Particle filters. Fig. 6b shows the error in the estimated path which is more than the error obtained while using the SDAE based Kalman filter and is not negligible. The estimation error is not negligible for both Kalman and Particle filters. The velocity of the moving obstacle estimated is given in Fig. 6c. The estimated velocity does not remain constant and produced oscillations. Comparing Figs. 5 and 6, it can be illustrated that the SDAE based Kalman filter outperforms the conventional Kalman filter and Particle filter by predicting the obstacle path and velocity more precisely. Table 2 summarizes a comparison of the performance of the developed prediction algorithm with that of the traditional Kalman filter, the particle filter, and the Kalman filter using DAE. As demonstrated in the table, the proposed method clearly outperforms existing methods [conventional Kalman filter, Particle filter, and Kalman filter using DAE]

(a) Integral squared error in prediction of X position

(b) Integral squared error in prediction of Y position

Fig. 7. Performance of the SDAE based extended Kalman filter (effect of noise)

in terms of ISE, IAE, and MAE. Since, the proposed SDAE based extended Kalman filter can predict an error free obstacle path, it can be used in applications like welding and drawing robots where a precise and error free estimated obstacle path is required. Initially, the weights of the SDAEs are randomly chosen. The encoder performance will not be satisfactory if the measured data consists of large noise. The weights can be optimized using Genetic algorithm and thereby the performance of the SDAE can be improved. Gaussian noises of different standard deviation such as 20%, 40% and 60% are added to the measured data. The measurement noise covariance is computed using the SDAE (i) with randomly chosen initial weights and (ii) with Genetic algorithm optimized weights. The computed measurement covariance matrix in both cases is used to predict the obstacle position. The integral squared error in the estimated x and y position in each case is shown in Fig. 7. The Kalman filter using SDAE with optimized weights has better performance as compared to the Kalman filter using SDAE with randomly chosen weights.

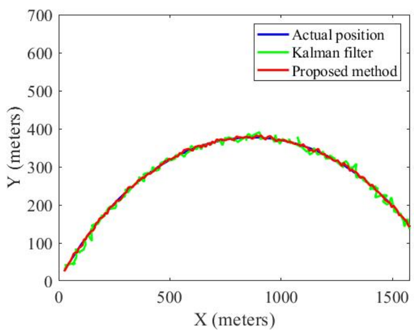
Table 2. Comparison of obstacle path prediction algorithms (linear motion)

Algorithm	ISE	IAE	MAE
Proposed method	0.421	0.212	0.023
Conventional Kalman filter	4.543	2.276	0.562
Particle filter	3.213	1.562	0.287
Kalman filter using DAE	0.496	0.295	0.031

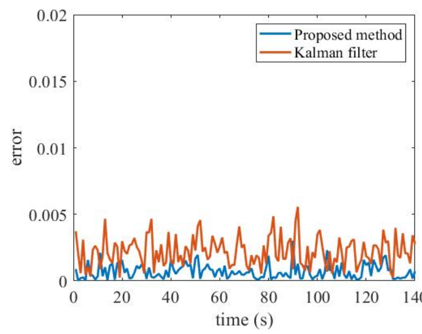
4.2.1 Obstacle with nonlinear path

Let the obstacle be a mobile robot car with state space model given by Equation (8), which moves along a nonlinear path. To evaluate the robustness of the developed algorithm the colored noise is added to the raw data. Pink noise, Brownian noise, and Azure noise are generated with inverse frequency power $\alpha = 1$, $\alpha = 2$, and $\alpha = -1$ respectively. The noisy measured data are given as inputs to the trained SDAEs which give noise free data as outputs. The measurement noise covariance matrix is determined using Algorithm 2 and is computed as $R = \begin{bmatrix} 0.21 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$.

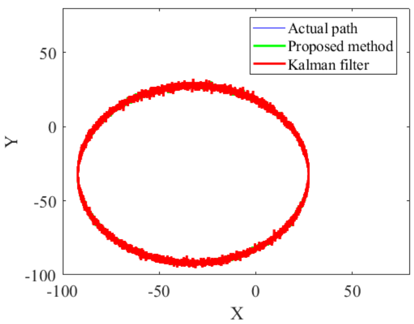
The measurement noise covariance matrix calculated is used for the computation of Kalman gain. The non linear path of obstacle is predicted using the SDAE based extended Kalman filter. Fig. 8a shows the estimated obstacle path using conventional extended Kalman filter and SDAE based extended Kalman filter. It is observed from this figure that the SDAE based extended Kalman filter is capable of estimating



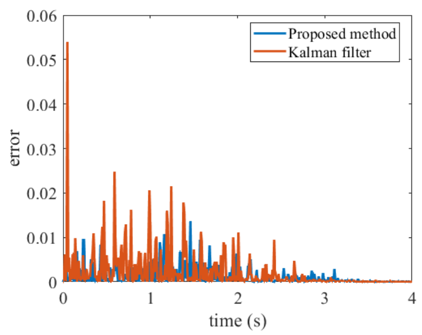
(a) Obstacle path estimated



(b) Error in estimated obstacle path



(c) Obstacle path estimated (circular path)

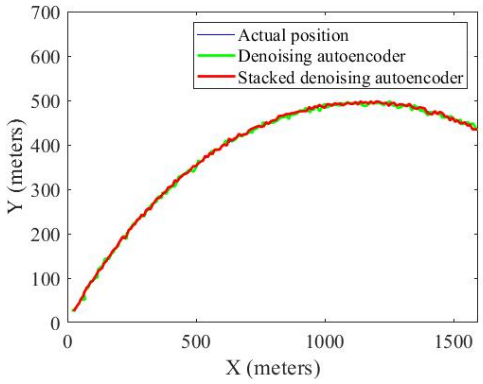


(d) Error in estimated obstacle path (circular path)

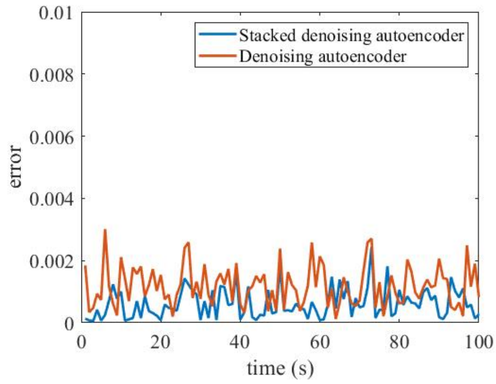
Fig. 8. Comparison of the SDAE based Kalman filter and conventional Kalman filter (nonlinear motion)

the nonlinear path more accurately as compared to the conventional extended Kalman filter. This observation is clearer from Fig. 8b which shows the estimated errors for the both methods. The estimated error is negligible for the proposed method. In Fig. 8c, the circular path predicted using both the traditional Kalman filter and the SDAE based Kalman filter is illustrated. The suggested technique has a higher estimation accuracy, as shown in Fig. 8d. Even though the error converges to zero in both cases, the conventional Kalman filter's maximum estimation error is substantial.

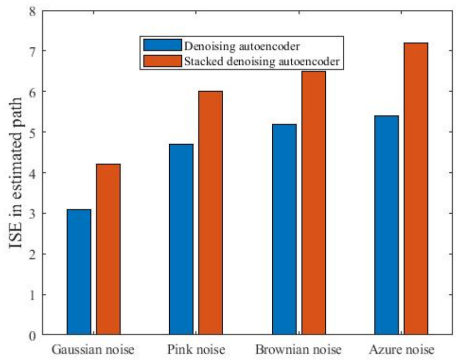
Neural network model with single layer fails to understand the training data set properly and produce results with error. More layers are added to extract more features from the data set. Thus, to produce an accurate output denoising autoencoder with stacked hidden layers are used. When SDAE and DAE are employed for determining the measurement noise covariance matrix R of the Kalman filter, the estimated nonlinear path and accompanying errors are shown in Figs. 9a and 9b, respectively. These figures demonstrate that the SDAE-based method produces the least amount of inaccuracy. To further understand the effectiveness of the proposed SDAE method, the integral squared error for both methods with Gaussian and the three colored noises are shown in Fig. 9c. In the presence of colored noise SDAE has better performance as compared to shallow neural network denoising autoencoder. The choosing of learning rate is one of the challenge in the stochastic gradient descent algorithm. Large learning rate results in the dwindling at minimum and small learning rate causes slow convergence. To increase the robustness of the stochastic gradient algorithm, Adam optimizer is used. The obstacle path is estimated using Kalman filter whose measurement noise covariance matrix are determined using SDAEs trained using both (i)Adam and (ii) stochastic gradient descent algorithms. During training both Gaussian noise and colored noise are considered. The performance of the proposed method with Adam and stochastic gradient descent learning algorithm is also analyzed which is shown in Fig. 10. A comparison of the performance of proposed method with existing algorithms in predicting the non linear motion of the obstacle is given in Table 3. The Adam optimizer has a better performance as compared to the stochastic gradient descent algorithm for both the colored and the Gaussian noises.



(a) Obstacle path



(b) Estimation error



(c) Performance comparison

Fig. 9. Comparison of stacked denoising autoencoder and denoising autoencoder

4.3 Performance of the proposed prediction algorithm in simulated environments

The performance of the proposed motion prediction algorithm is quantitatively tested in MATLAB simulated environments. In the simulation scenario 1, a dynamic environment with three moving obstacles shown in Fig. 11 is considered. Let the start position of the robot be (0,0) and the goal position be (12,10). Initially, the path is planned offline considering that the obstacles are static. The moving obstacles are detected using ultrasonic sensor. Once the dynamic obstacles are detected, the obstacle path has to be estimated to ensure collision free navigation. The obstacle path is predicted using the Kalman filter where the Kalman gain is calculated using Equation (12) for which the measurement noise covariance matrix is to be determined. The measurement noise covariance matrix is computed using Equation (15) and is obtained as $R = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}$.

The estimated obstacle path is compared with the robot path planned initially. From Fig. 11, it is clear that the initially planned path collides with the obstacle path so the path is to be re-planned. Thus, an optimal and collision free path is obtained. The uncertainty in prediction of the obstacle path using both the Kalman filter and the SDAE based Kalman filter is depicted in Fig. 12. The uncertainty in obstacle path prediction is large for the conventional Kalman filter which will affect the robot navigation in applications that require precise path.

4.3.1 Comparison of the performance of the proposed algorithm

To evaluate the efficacy of the proposed path planning algorithm using SDAE based extended Kalman filter, the proposed method is compared with that of (i) hybrid A star (ii) artificial potential field (iii) dynamic path planning using decision algorithm. The path length, computation time, and the ability to obtain collision free path in closely spaced obstacles are considered here for evaluation. The computation

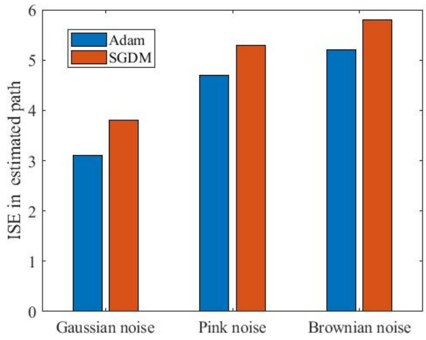


Fig. 10. Comparison of Adam and SGDM

Table 3. Comparison of obstacle path prediction algorithms (non-linear motion)

Algorithm	ISE	IAE	MAE
Proposed method (Adam optimizer)	0.534	0.158	0.021
Proposed method (Stochastic method)	0.942	0.382	0.043
Conventional Kalman filter	3.573	1.416	0.328
Kalman filter using DAE	1.32	0.4382	0.064

time is obtained using MATLAB 2020a. A MATLAB simulation environment is considered with both static and dynamic obstacles (scenario 2). The initial position of the robot is (8,0) and the goal position is (10,10). The proposed path planning algorithm estimates the obstacle path using SDAEs based extended Kalman filter whereas in the hybrid A star method, the obstacle motion is assumed to follow a constant velocity. The robot path planned using the proposed algorithm is shown in Fig. 13a. The initial planned path collides with the obstacle path and is re-planned. The path obtained using hybrid A star algorithm is given in Fig. 13b. The hybrid A star algorithm calculate the cost function at each node and finds the optimal path. Comparing Figs. 13a and 13b, it can be elucidated that the proposed path planning algorithm is able to find the shortest and optimal path from the initial position to final position and thus, the proposed algorithm outperforms the hybrid A star path planning algorithm. The path achieved by the potential field algorithm in the dynamic environment is shown in Fig. 13c. The dynamic obstacle is having a random motion and is shown in Fig. 13c. The potential field algorithm fails to achieve a collision free path when the obstacles are closely packed. The proposed algorithm finds the shortest and collision free path from the start position to the goal position when compared to hybrid A star and

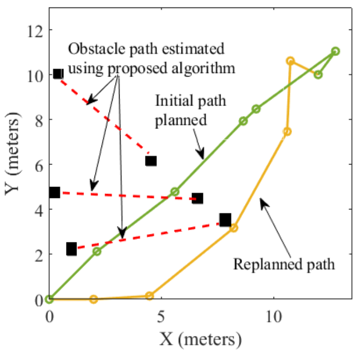


Fig. 11. Path planning (scenario 1)

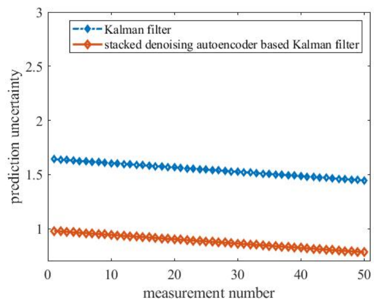
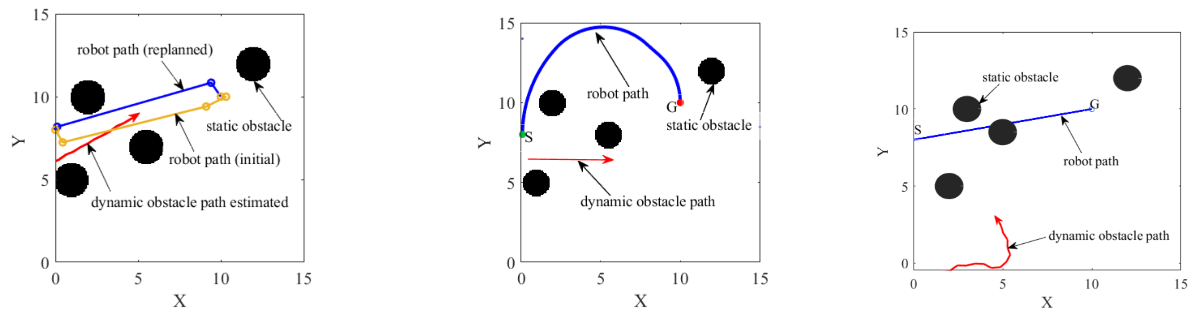
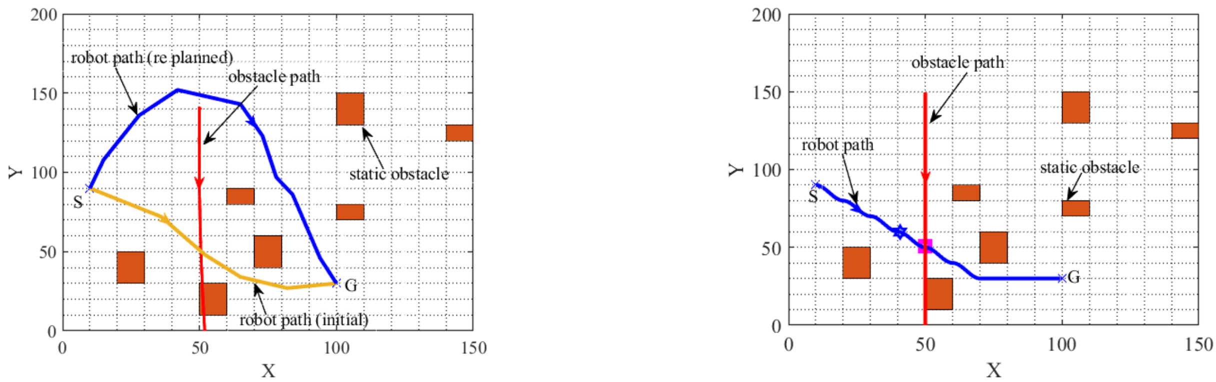


Fig. 12. Uncertainty in prediction



(a) Proposed algorithm **(b)** Hybrid A star **(c)** Artificial potential field

Fig. 13. Comparison of performance of proposed algorithm (scenario 2)



(a) Path planning using proposed algorithm **(b)** Path planning using decision algorithm

Fig. 14. Comparison of performance of proposed algorithm (scenario 3)

artificial potential filed algorithms.
The suggested algorithm is compared to the decision algorithm (Xidias, 2021), which takes both dynamic and static impediments into account. When the obstacle enters the threshold domain, the robot's velocity is reduced, and the robot must wait until the obstacle departs the threshold region, according to the decision algorithm. When the distance between the obstacle and robot exceeds the threshold value, the robot's velocity is boosted, allowing it to approach the goal. The path planning in scenario 3 using the decision algorithm is depicted in Fig. 14b. The robot must wait till the obstruction has passed, resulting in a longer computation time. The presented algorithm, as shown in Fig. 14a, re-plans the robot path when there is a collision between the obstacle and the robot path. The computation time in each of the algorithms is computed using MATLAB 2020a. The computation time is minimum for the proposed algorithm while compared to decision algorithm. In Table 4, a comparison of the suggested algorithm with the existing path planning algorithms is given. Analyzing the simulation results, it can be concluded that the SDAE based extended Kalman filter with Adam optimizer predict the obstacle path precisely. The proposed algorithm produced negligible error in the presence of both colored (brown, pink, and azure) and white noise. Also, the prediction uncertainty is less for the proposed algorithm which is a key factor in robot navigation. By accurately predicting the obstacle motion, the robot is able to achieve a collision free navigation in the dynamic environment. The developed algorithm outperforms the conventional Kalman filter and the denoising based extended Kalman filter. In comparison to the (Sedighi S., *et al.*, 2019), (Ge S.S., *et al.*, 2002), and (Xidias, 2021), path planning employing the developed methodology is faster and more robust in narrow passages.

Table 4. Comparison of path planning algorithms (dynamic environment)

Algorithm	Computation time (s)	Robustness in narrow passages
Proposed method (scenario 3)	104.643	yes
Decision algorithm (scenario 3)	247.867	yes
Proposed method (scenario 2)	64.342	yes
Hybrid A star (scenario 2)	78.249	yes
Artificial potential field (scenario 2)	68.214	no

5. Conclusion

A SDAE-based extended Kalman filter is proposed in this paper for predicting obstacle motion in dynamic scenarios. The SDAE is a deep neural network whose input is a noisy sensor data and output is the noise free data. The noisy and noise free data is used to get the measurement noise covariance matrix of the extended Kalman filter which is used to determine the path of a moving obstacle. To train the neural network, a set of noise free data are collected which are considered as the targets for the training purpose. The input of the SDAE during training is obtained by adding noises to the target data. Once the SDAE is trained then it can give the optimum measurement covariance matrix. The SDAE is capable of effectively denoising the measured data in the presence of both Gaussian noise and colored noise. MATLAB simulations are carried to predict the path of moving obstacle with conventional extended Kalman filter, Particle filter and by using the proposed SDAE based extended Kalman filter. The results illustrated that the extended Kalman filter using the SDAE gives a much accurate path for both linear and nonlinear obstacle paths. The simulation study also illustrated that the ISE, IAE, and MAE in the estimated obstacle path is very less with the SDAE based extended Kalman filter whose learning algorithm is Adam. But the training time is more for an Adam optimizer while compared to stochastic descent algorithm. Different scenarios are considered in MATLAB simulations to test the effectiveness of the proposed method for determining the optimal path in a dynamic environment with multiple impediments. Using MATLAB simulated testing environments, the performance of the proposed method in path planning is compared against hybrid A star, artificial potential field, and decision algorithms. The suggested methodology achieves an optimal collision-free path with minimal computing time in various testing scenarios.

ACKNOWLEDGMENT

This work was supported by All India Council for Technical Education-National Doctoral Fellowship (NDF-RPS) scheme.

References

Ariff, M.A.M., 2021. A new intelligent time-series prediction technique for coherency identification performance enhancement. Kuwait Journal of Science, 48(4).

Berntorp, K. & **Di Cairano, S., 2016, July.** Particle filtering for online motion planning with task specifications. In 2016 American Control Conference (ACC) (pp. 2123-2128). IEEE

Chen, J., Zhou, Y., Gong, J. & **Deng, Y., 2019, July.** An improved probabilistic roadmap algorithm with potential field function for path planning of quadrotor. In 2019 Chinese Control Conference (CCC) (pp. 3248-3253). IEEE.

Diversi, R., Guidorzi, R. & **Soverini, U., 2005.** Kalman filtering in extended noise environments. IEEE Transactions on Automatic Control, 50(9), pp.1396-1402.

Elnagar, A. 2001, July. Prediction of moving objects in dynamic environments using Kalman filters. In Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No. 01EX515) (pp. 414-419). IEEE.

Gao, J., He, Q., Gao, H., Zhan, Z. and Wu, Z., 2018. Design of an efficient multi-objective recognition approach for 8-ball billiards vision system. Kuwait Journal of Science 45.1 (2018).

Ge, S.S. & **Cui, Y.J., 2002.** Dynamic motion planning for mobile robots using potential field method. Autonomous robots, 13(3), pp.207-222.

Khan, M.S.A., Hussian, D., Ali, Y., Rehman, F.U., Aqeel, A.B. and Khan, U.S., 2021, November. Multi-Sensor SLAM for efficient Navigation of a Mobile Robot. In 2021 4th International Conference on Computing & Information Sciences (ICCIS) (pp. 1-5). IEEE.

Lin, X., Wang, Z.Q. & **Chen, X.Y., 2020, May.** Path Planning with Improved Artificial Potential Field Method Based on Decision Tree. In 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS) (pp. 1-5). IEEE.

Lin, Y. & **Saripalli, S., 2017.** Sampling-based path planning for UAV collision avoidance. IEEE Transactions on Intelligent Transportation Systems, 18(11), pp.3179-3192.

Liu, Z., Jiang, Z., Xu, T., Cheng, H., Xie, Z. & **Lin, L., 2018, May.** Avoidance of high-speed obstacles based on velocity obstacles. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7624-7630). IEEE.

Matisko, P. & **Havlena, V., 2010.**Noise covariances estimation for Kalman filter tuning. IFAC Proceedings Volumes, 43(10), pp.31-36.

Mehra, R. 1970. On the identification of variances and adaptive Kalman filtering. IEEE Transactions on automatic control, 15(2), pp.175-184.

Odelson, B.J., Rajamani, M.R. & **Rawlings, J.B., 2006.** A new autocovariance least-squares method for estimating noise covariances. Automatica, 42(2), pp.303-308.

Park, J.S. & Manocha, D., 2020. HMPO: human motion prediction in occluded environments for safe motion planning. arXiv preprint arXiv:2006.00424.

Park, S., Gil, M.S., Im, H. & **Moon, Y.S., 2019.** Measurement noise recommendation for efficient Kalman filtering over a large amount of sensor data. Sensors, 19(5), p.1168.

Prevost, C.G., Desbiens, A. & **Gagnon, E., 2007, July.** Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In 2007 American control conference (pp. 1805-1810). IEEE.

Ren, Z., Lai, J., Wu, Z. and Xie, S., 2021. Deep neural networks-based real-time optimal navigation for an automatic guided vehicle with static and dynamic obstacles. Neurocomputing, 443, pp.329-344.

Roggeman, H., Marzat, J., Derome, M., Sanfourche, M., Eudes, A. & **Le Besnerais, G., 2017.** Detection, estimation and avoidance of mobile objects using stereo-vision and model predictive control. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 2090-2099).

Saricicek, I., Keser, S.B., Cibi, A. and Ozdemir, T., 2022. Energy Efficient Routing and Task Scheduling for Autonomous Transport Vehicles in Intra Logistics. Kuwait Journal of Science, 49(1).

Sedighi, S., Nguyen, D.V. &**Kuhnert, K.D., 2019, April.** Guided hybrid A-star path planning algorithm for valet parking applications. In 2019 5th international conference on control, automation and robotics (ICCAR) (pp. 570-575). IEEE.

Shumway, R.H. & **Stoffer, D.S., 2019.** Time series: a data analysis approach using R. Chapman and Hall/CRC.

Soydaner, D., 2020 A comparison of optimization algorithms for deep learning. International Journal of Pattern Recognition and Artificial Intelligence, 34(13), p.2052013.

Valappil, J. & **Georgakis, C., 2000.** Systematic estimation of state noise statistics for extended Kalman filters. AIChE Journal, 46(2), pp.292-308.

Van Den Berg, J.P. & **Overmars, M.H., 2005.** Roadmap-based motion planning in dynamic environments. IEEE Transactions on Robotics, 21(5), pp.885-897.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A. & **Bottou, L., 2010.** Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12).

Völz, A. & **Graichen, K., 2019.** A predictive path-following controller for continuous replanning with dynamic roadmaps. IEEE Robotics and Automation Letters, 4(4), pp.3963-3970.

Wang, S.L. 2013. Research on key technology of multi-GNSS ground based augmentation system. Southeast Univ., Nanjing, China, Tech. Rep, pp.13-27.

Wei, H., Huang, Y., Hu, F., Zhao, B., Guo, Z. and Zhang, R., 2021. Motion Estimation Using Region-Level Segmentation and Extended Kalman Filter for Autonomous Driving. Remote Sensing, 13(9), p.1828.

Wu, F., Luo, H., Jia, H., Zhao, F., Xiao, Y. & **Gao, X., 2020.** Predicting the Noise Covariance With a Multitask Learning Model for Kalman Filter-Based GNSS/INS Integrated Navigation. IEEE Transactions on Instrumentation and Measurement, 70, pp.1-13.

Xidias, E.K., 2021. A Decision Algorithm for Motion Planning of Car-Like Robots in Dynamic Environments. Cybernetics and Systems, pp.1-20.

Xing, C., Ma, L. & **Yang, X., 2016.** Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. Journal of Sensors, 2016.

Yayan, U., Yazici, A. and Saricicek, I., 2021. Prognostics-aware multi-robot route planning to extend the lifetime. Kuwait Journal of Science.

Yuen, K.V., Liang, P.F. & **Kuok, S.C., 2013.** Online estimation of noise parameters for Kalman filter. Struct. Eng. Mech, 47(3), pp.361-381.

Zhu, Q., Han Y., Liu, P., Xiao, Y., Lu, P. and Cai, C., 2019. Motion planning of autonomous mobile robot using recurrent fuzzy neural network trained by extended Kalman filter. Computational intelligence and neuroscience, 2019.

Submitted: 29/01/2022
Revised: 30/03/2022
Accepted: 03/04/2022
DOI: 10.48129/kjs.18361