ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 12 ||December – 2015 ||
International Journal of Computational Engineering Research (IJCER)

A Novel Algorithm for Autonomous Robot Navigation System Using Neural Network

Najmuddin Aamer[1], S. Ramachandran[2]
[(1)]Research Scholar, V.M.U, Salem, Tamil Nadu, India
[(2)] S.J.B. Institute of Technology, Bengaluru, India
[1]najmuddinaamer@gmail.com, [2]ramachandr@gmail.com

Abstract:
Autonomous robot is a type of the robot which can perform operation accurately in any environment. Currently, the demands of robots are increasing in the industry; the autonomous robots suffer from the path planning and obstacle avoidance. To overcome this, many approaches have been proposed but due the noisy environment, it is difficult for the robots to perform navigation. In this work, we proposed an efficient model for the robot navigation using neural networks and to overcome the path planning, kinematics based model is presented. By utilizing the kinematics based stability model, we are able to achieve the stability condition of the robot and it is used for the controlling the speed of the robot. Neural network is used for the obstacle prediction and avoidance. The simulation result presented that the proposed model is robust to noisy (hindrance) environment and it is able to avoid the obstacle precisely at the maximum speed of 0.02 to 1.0 m/s.

Keywords: *Autonomous Robot Navigation, Path Planning, Obstacle Avoidance, Neural Network, Kinematics based model.*

I. Introduction

Recently some approaches have been proposed based on the machine learning, to help enhance the capability of the mobile robot system. This method is useful for the autonomous robot control and navigation system. Autonomous robot is the type of robot which performs the operation precisely. It is performed using autonomous control system. To perform the operation the main point is to planning the path. Navigation of robot is the main issue in robot a system which includes achieving target along with avoiding the obstacle [1]. It is essential for any robot to understand the structure of the deployment. Robots must have some decision making technique, data preprocessing, path planning etc. Amid the improvement of new route algorithm, it is important to test them in robots and situations before the testing on genuine robots and this present reality. This is on the grounds that (i) the costs of robots are broad; (ii) the untested calculation may harm the robot amid the investigation; (iii) challenges on the development and shift of framework models under clamor foundation; (iv) the transient state is hard to track decisively; and (v) the estimations to the outside signals are covered up amid the trial, however this data is frequently useful for troubleshooting and upgrading the calculations.

The product test system could be a decent answer for these issues. A decent test system could give various situations to encourage the specialists to discover issues in their calculations in various types of versatile robots. Keeping in mind the end goal to take care of the issues recorded over, this test system should have the capacity to screen framework states nearly. It additionally ought to have adaptable and amicable clients' interface to build up a wide range of calculations. Many approaches have been proposed for mobile robot navigation using intelligent computations i.e. Particle Swarm Optimization (PSO), Fuzzy Logic(FL) [2, 3]. The key challenge in autonomous robot navigation is robust planning of path and avoiding the obstacles. The state of the art methods are inefficient for the avoiding the obstacle, so to eliminate this problem a promising method is required for the efficient navigation of mobile robots. Path planning can be divided into two parts (i) Global method of path planning and (ii) Local path planning method. According to the former method optimal path can be achieved by utilization of off-line computations, the main drawback of this method is that it is not able to adapt the variation in the environment. Later method of path planning is known as most adaptable method which can adapt the environment conditions without any supervision. The main approach proposed in this field is Simultaneous Localization and Map Building (SLAM) algorithm. This algorithm uses the features of the map to for estimation the locations of robot and obstacle.

Path planning is central assignment of the versatile robots routes, where the precision of way relies on the ecological mapping and confinement. A few path planning methodologies are now utilized for precise way arranging, for example, Dijkastra's Algorithm, Visibility Graphs, Cell Decomposition Technique, A* and Modified A* Algorithms and so forth. A* system does not support the exact way arranging if the robot's span is bigger than the cell's measure. In such circumstances it is hard to move the portable robot through limited entryway or section [18]. One of the key issues in mobile robot systems is real time path planning. Autonomous navigation involves obstacle avoidance along with moving towards a target. There have been several approaches towards this issue, encompassing both the static and dynamic environment problem. One approach to autonomous navigation is the wall following method [1, 2]. Here the robot navigation is based on moving alongside walls at a predefined distance, while considering the obstacles as just another wall. Though computationally less demanding, this approach has only specific applications (a floor cleaning robot in a long hallway). An improvement upon this method is the edge detection approach [3, 4], where the positions of the vertical edges of the obstacle are determined and the robot is steered around either edge. However this approach is heavily dependent upon sensor accuracy. A powerful scheme for autonomous navigation is the Potential Fields Method, originally suggested by Khatib [5]. Here obstacles are considered as centers of repulsion and targets as centers of attraction (with global effect). The robot traverses the path of least potential gradient. A shortcoming of this method is that it assumes a known and prescribed world model of obstacles. Later a model called the Virtual Force Field Model [6] was developed which employed the integration of the above mentioned concept and the concept of Certainty Grid.

Route for versatile robots is firmly identified with sensor-based way arranging in 2D, and can be considered as a developed zone of exploration [26, 27]. Sensor-based getting ready for controllers, then again, is still an extremely open exploration issue. One of the first frameworks fit for sensor based getting ready for control was the Handy framework [28, 29]. Based on laser extent information, this framework could plan pick-and-place undertakings for a controller furnished with a parallel-jaw gripper. For two late cases, see Ahuactzin and Portilla [30].

There are always static, as well as non-static obstacles in the environment. Hence, robots need to autonomously navigate themselves in environments by avoiding obstacles. The neural networks, which have been designed for obstacle avoidance by mobile robots, should take the sensor data from the environment as their inputs, and output the direction or the robot to Proceed. In Fujii et al. [31] presented a multilayered neural network model through reinforcement learning for collision avoidance of a mobile robot. Silva et al. proposed the MONODA (Modular Network for Obstacle Detection and Avoidance) architecture for obstacle avoidance and detection of a mobile robot in unknown environments [8]. This model consists of four three-layered feed forward neural network modules where each module detects the probability of obstacles in one direction of the robot. In Ishii et al. [32] developed an obstacle avoidance method based on self organizing Kohonen neural networks for underwater vehicles. Gaudiano and Chang proposed an approach for obstacle avoidance by employing a neural network model of classical and operant conditioning based on Grossberg's conditioning circuit [7, 33]. Parhi and Singh introduced a real-time obstacle avoidance approach to solve each of the target seeking, obstacle-avoidance, and wall-following tasks using separate neural networks [19, 20, 21]. In their approach, based on certain criteria one of the networks is selected at each time step to control the mobile robot allowing it to move safely in a crowded real world and unknown environment and to reach a specified target while avoiding static as well as dynamic obstacles.

This paper manages the utilization of Neural Networks in independent route. Neural Network methodologies have been utilized as a part of very much a couple way planning algorithm [7], where a Hopfield kind of system with constant neurons was utilized. In this paper we have utilized a model approximately in view of a Competitive Learning model for deterrent allowed to find an objective moving in a subjective manner. Unlike most methodologies, this strategy does not require the worldwide picture, every neuron in the neural system having just neighborhood associations. The always upgrading neural exercises produce a movement scene which controls the robot towards its objective. To find the obstacle free direction, a multi-layer method is utilized in this approach. In this paper we have worked on the both aspects of path planning and obstacle avoidance. The first stage is to planning the path and then if any obstacle experienced then avoiding the obstacle. But main concentration in this paper is about planning the path which is the crucial stage for any navigation system.

The rest of the paper is organized as follows. Related work in this manuscript is explained in Section II. Section III describes the proposed simulator. Experimental results are given in Section IV to show the system performance. Finally, Section V presents a brief conclusion and potential future work

II. Related work

As the earlier discussion presents that the main aspect in autonomous robot is avoiding the obstacle. In this context an overview about the path planning is presented in some existing methods. Reasoning and sensing are the main objectives for navigation of robots. An efficient reasoning method using Finite State Machine (FSM) was proposed by Qingxiang Wul et al. [6]. Evaluation of the system performance Khepera robot was used. This method demonstrates the human knowledge transfer to the autonomous robot [6]. A new method for robot navigation using bottom – up approach is proposed which is cooperative intelligent system. The method is divided into two subparts; intelligent devices based on microcomputer and intelligent behavior method for robot navigation [8].

Neural network is an intelligent method for prediction of the data. Neural network also has been used for robot navigation. Recurrent neural network is mainly used for the control of the robot. Four architecture of recurrent neural network are presented in [9].

In order to resolve the problem of robot navigation challenging tasks are such as optimization, adaption, and generalization, decision making and learning. Fuzzy logic and neural networks are promising methods to resolve these issues. The main advantages of these algorithms are the adaptation of the environment and learning. These methods are robust to the noisy environment [10].

Pinto A.B. et al. [11] applied the ideas connected with neural systems in the control of an independent robot. The robot stage utilized empowered testing and acceptance of the ANN control and its correlation with other broadly utilized Procedures, for example, FLC. For the processing requirement of test information from the robot, the stage was hallowed with a remote control framework created in an Android Smartphone. Two methodologies were taken in actualizing the robot control. In the first approach, the robot was taught to perform like the control by fuzzy logic that was already created. The second approach taught the robot from information taken from the remote control framework. Both methodologies were at first actualized and simulated in MATLAB, before testing and executing the calculations in the robot stage.

An assessment of expense productive, lightweight, less loud and low upkeep mechanical framework was proposed by Hassan K.M. et al. [11]. At the same time, having the facility of programmed shirking of any hindrances and equipped for discovering its way around. Then again, the robot may perform floor cleaning capacities like vacuum cleaning and clearing while moving around. Mapping innovation is truant in this configuration because of practical issue. The thought make inclination for the robot to clean the whole region of a room. Two or three turning side brushes are appended to the underneath of the cleaning machine to collect soil, flotsam and jetsam including pet-hair also amid the move along the way. Despite the fact that the robot is round formed yet it can in any case clean along edges and into other difficult to-achieve places. The robot can likewise be moved utilizing remote control to achieve the destination [12].

An imaginative versatile mechanical framework intended for surveillance systems was proposed by Maykol Pinto et al. [13]. This versatile robot moves along a rail and is furnished with a monocular camera. In this manner, it upgrades the reconnaissance ability when contrasted with customary frameworks (for the most part made by various static cameras). Also, the paper proposes a method for multi-article following called MTMP (Multi-Tracking of Motion Profiles). The MTMP resorts to a plan in view of the Kalman channel and tracks a few moving items utilizing movement profiles. A movement profile is portrayed by the overwhelming stream vector and is registered utilizing the optical stream signature with evacuation of exceptions. A likeness measure taking into account the Mahalanobis separation is utilized by the MTMP for partner the moving items over edges.

The previous couple of decades have seen a ton of examination movement in the field of versatile mechanical technology which includes the conduct of robots under dynamic and testing conditions to accomplish a certain objective. The traditional methods like perceivability diagrams, Cell Decomposition, and Potential fields have been generally utilized as a part of movement arranging. These traditional systems however experience the ill effects of a few weaknesses like high time intricacy in high measurements and issues of getting caught in neighborhood minima. Delicate figuring Proceedings endures like fuzzy logic and other organically motivated systems like simulated neural systems, genetic algorithm, ant colony and different strategies have likewise been broadly utilized for robot movement arranging [14, 15, 16].

A route framework in view of Support Vector Machine (SVM) way arranging and fuzzy Sliding-Mode Controlled (SMC) way devotee is created for wheeled specialists in [17]. The created framework, involving image acquiring, guide designing, way arranging, mark task and way following derivation component, means to effortlessly take after an arranged smooth way. In the first partition, a Voronoi chart is utilized as a preprocessor to generally fit a sheltered course between the introductory and objective positions with irregular fragmental sections to maintain a strategic distance from deterrents. A postprocessor of Gaussian bit SVM is then connected to sequentially change over the fragmental way to a differentiable consistent way for a sheltered and smooth entry logged off. In the second section, a way following derivation component in light of fuzzy SMC is proposed thusly to track the arranged way on line [17].

Yunfei Zhang et al. [19] added to a various leveled controller to dodge haphazardly moving deterrents in independent route of a robot. The created technique comprises of two sections: an abnormal state Q-learning controller for picking an ideal arrangement for route and a low-level, Appearance-Based Visual Servo (ABVS) controller for movement execution. The utilization of robot learning capacity in crash evasion is a novel component, in a joined framework system of arranging and visual servo control. The created methodology exploits the on-board camera of robot whose limited field of perspective is normally suitable for the Q-learning calculation. In view of the Q-learning controller, information of deterrent development and a control law for the ABVS controller are not required. This is a critical computational point of preference. The technique is actualized in a recreation arrangement of robot route.

Becerra, H. M. et al. [20] proposed a novel control law taking into account sliding-mode hypothesis so as to drive portable robots to an objective area, which is indicated by a formerly obtained reference image. The control plan exploits the piecewise epipolar geometry of three perspectives on the premise of picture based visual servoing, in a manner that no 3-D scene data is required. The paper's commitment is another control law that accomplishes meeting to the objective with no assistant pictures and without changing to any methodology other than epipolar-based control. Moreover, the utilization of sliding-mode control manages singularities, in this manner permitting the robot to move straightforwardly toward the objective and also maintaining a strategic distance from the need of an exact camera alignment.

Lapierre, L., Zapata et al. [21] proposed a method to drive unicycle robot to follow the desired path, it handles obstacle avoidance capacity also. The path following control configuration depends on Lyapunov hypothesis, back stepping strategies and arrangements unequivocally with vehicle motion. Moreover, it overcomes starting condition requirement present in various way taking after control methodologies depicted in the literature.

Amato et al. [22] propose that, for samples that are not collision free, binary search along random directions can be used for quickly generating configurations near the boundary of Cobs. Thus, colliding configurations act as seeds for configurations near the C-space obstacles.

Gilbert and Hong [23] formulated the collision detection problem as a root finding problem, where the root (if it exists) is the first collision on the given path of motion. Another technique is to check for collision between the volumes that are swept out by the geometric objects as they move.

III. Proposed model

In existing works we have discussed about the various approaches for controlling the autonomous robot and the navigation. In this section we present a robust approach for the efficient navigation of mobile robot navigation.

Artificial neural networks are information-processing systems which have certain performance characteristics in common with biological neural networks [22]. Artificial neural networks have been evolved as generalizations of mathematical models of human cognition or neural biology, based on the following four assumptions [17]:

1. "Information processing occurs at many simple elements called neurons."
2. "Signals are passed between neurons over connection links."
3. "Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted."
4. "Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal."

A neural network can be characterized, firstly, by its structure of connections between the neurons (known as its architecture), additionally by its method of determining the weights on the connections (called its training, or learning, algorithm), and finally, its activation function.

Table 1. Neural network activation functions [17, 22]

Sl. No.	Function	Definition	Range
a)	Identity	x	$(\infty, +\infty)$
b)	Binary Sigmoid	$\frac{1}{1+e^{-x}}$	$(-1, +1)$
c)	Bipolar Sigmoid	$\frac{1-e^{-x}}{1+e^{-x}}$	$(-1, +1)$
d)	Hyperbolic	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$(0, +1)$
e)	Exponential	e^{-x}	$(-1, +1)$
f)	Softmax	$\frac{e^{x}}{\sum_i x_i}$	$(0, +\infty)$
g)	Unit Sum	$\frac{x}{\sum_i x_i}$	$(0, +1)$
h)	Square Root	$\sqrt{x}$	$(0, +\infty)$
i)	Sine	$Sin(x)$	[0,+1]
j)	Ramp	$\left\{\begin{matrix} -1 \\ 0 \quad < x \\ +1 \end{matrix}\right.$ $\left.\begin{matrix} x \leq -1 \\ < \quad x < 1 \\ x \geq 1 \end{matrix}\right\}$	[-1,+1]
k)	Step	$\left\{\begin{matrix} 0 & x < 0 \\ +1 & x \geq 0 \end{matrix}\right\}$	[0,+1]

Essentially, neural network deals with cognitive tasks such as learning, adaptation, generalization and optimization. Indeed, recognition, learning, decision-making and action constitute the principal navigation problems. To solve these problems fuzzy logic and neural networks are used. A neural network is a massive system of parallel distributed processing elements (neurons) connected in graph topology. Learning in the neural network can be supervised or unsupervised. Supervised learning uses classified pattern information, while unsupervised learning uses only minimum information without reclassification. Unsupervised learning algorithms offer less computational complexity and less accuracy than supervised learning algorithms. Then, former learn rapidly, often on a single pass of noisy data. For the obstacle avoidance purposes recurrent type of neural network was used with the gradient back propagation technique for training the network. The using of supervised neural network for robot navigation is in partially known in the environment.

a. CONTROL SYSTEM OF MOBILE ROBOT

In the previous section we have analyzed that for robot navigations a control system is required. This section presents the proposed model of control system which is shown in Figure 1.

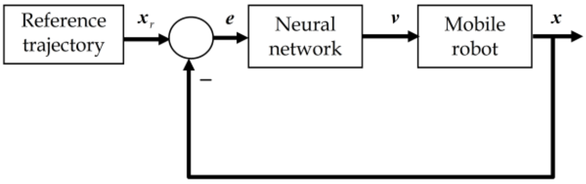


Figure 1. Mobile robot motion control system

The above given Figure presents the overall controller model for the proposed autonomous robot navigation using neural network. For proposed system the main input parameter are the user defined trajectories or co-ordinate points. Based on these points neural network prediction is performed which allows robot to make the movements in the predicted directions.

Let us consider an autonomous mobile robot which consist two wheels which are kept on the same axis. The motion is provided to the wheels by using actuators. For initial localization of robot position of robot is defined as (x, o, y) and the center of mass is defined as c.

Based on prediction to assign the movement, robots require some kinematics parameters which can be defined as shown in equation (1).

$$\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix} = \begin{bmatrix}\cos(\gamma) & 0\\ \sin(\gamma) & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}\psi\\ \omega\end{bmatrix} \tag{1}$$

Where $\dot{\alpha}$ and $\dot{\beta}$ presents the initial localization coordinates of the mobile robots, orientation angle of the mobile robot is given by γ since vechicle is having orientation angle, so angular and liner velocities also present in the model which is presented as ω and ψ respectively.

Mobile robot contains two moving wheels so for each wheel velocity is defined as ψ_R (Right wheel linear velocity) and ψ_L (Left wheel velocity).

$$\begin{bmatrix}\psi\\ \omega\end{bmatrix} = \begin{bmatrix}\xi & \xi\\ \dfrac{\xi}{B_{path}} & \dfrac{\xi}{B_{path}}\end{bmatrix}\begin{bmatrix}\psi_R\\ \psi_L\end{bmatrix} \tag{2}$$

Where ξ represents the diameter of the mobile robot wheel, B_{path} represents the path of the robot, ψ_R (Right wheel linear velocity) and ψ_L (Left wheel velocity).

Combining the equation (1) and (2)

$$\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix} = \begin{bmatrix}\xi\cos(\gamma) & \xi\cos(\gamma)\\ \xi\sin(\gamma) & \xi\sin(\gamma)\\ \dfrac{\xi}{B_{path}} & -\dfrac{\xi}{B_{path}}\end{bmatrix}\begin{bmatrix}\psi_R\\ \psi_L\end{bmatrix} \tag{3}$$

By using these kinematics equation robot's stability condition can be achieved for the robot which is shown in Figure 2

$$\alpha \sin(\gamma) + \dot{\beta} - d\dot{\gamma} = 0 \qquad (4)$$

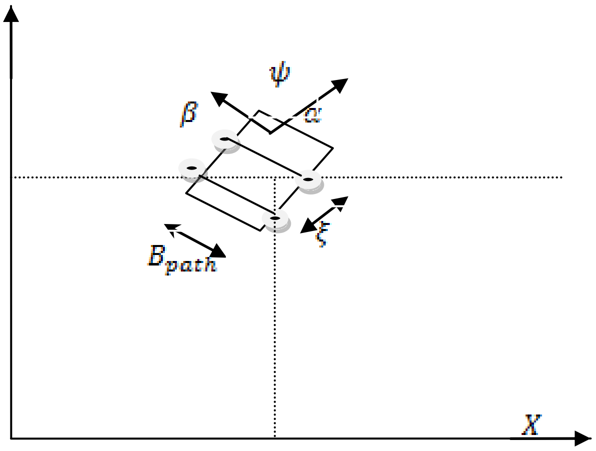


Figure 2. Mobile Robot Presentation

In the below given Figure 3 the overall navigation flow chart of the proposed model is presented. Initially when the robot is initialized, Two coordinates are given as input : (i) Robot location coordinates, (ii) Target Coordinates. Based on the target coordinates the distance is computed and measured whether the computed distance is correct or not? If it is correct then the robot will start moving towards target point else again the distance will computed and measured. After this step, when the robot is moving towards the target it keeps on checking the location to match with the final destination. If it matches with the target points then simulation is stopped or else again it starts moving towards the target.

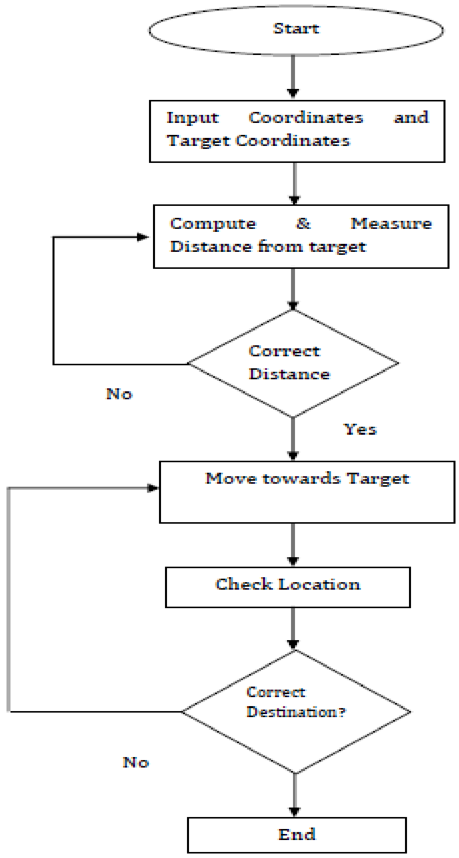


Figure 3. Robot Navigation Algorithm Flow

In order to plan the path and train the neural network the procedure is shown in Figure 4.

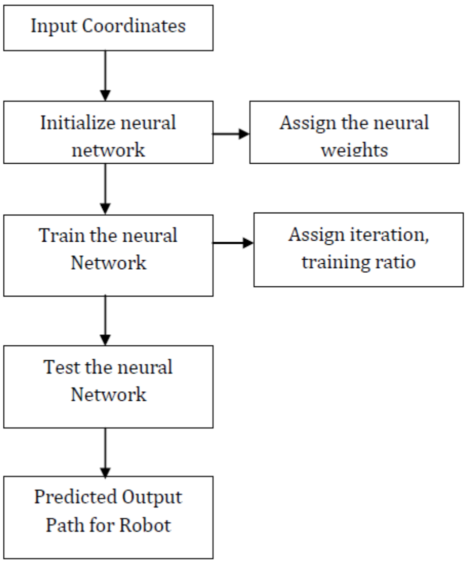


Figure 4. Overall Architecture of the Proposed Model

b. PROPOSED MODEL FOR NEURAL NETWORK CONTROLLER

Analysis presented in the previous section gives the stability condition for the motion of the robot. Now the next step is to design robust model for the neural network controller.

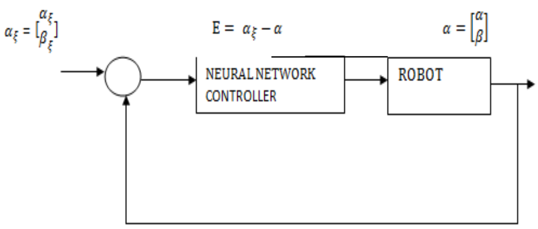
In order to design a neural network controller the trajectory arc need to be sensed by the robot while moving which can be defined as

$$\tau = \frac{\gamma\xi}{\sin(\gamma)} \tag{5}$$

Where τ is the trajectory arc of the robot

For this arc trajectory of robot the time domain derivative can be estimated as

$$\begin{bmatrix} \frac{d\alpha}{dt} \\ \frac{d\beta}{dt} \end{bmatrix} = \begin{bmatrix} \xi\cos(\gamma) & \xi\cos(\gamma) \\ \xi\sin(\gamma) & \xi\sin(\gamma) \\ \frac{\xi}{B_{path}} & -\frac{\xi}{B_{path}} \end{bmatrix} \begin{bmatrix} \psi_R \\ \psi_L \end{bmatrix} \tag{6}$$



After rewriting equation (6)

$$\begin{bmatrix} \frac{d\alpha}{dt} \\ \frac{d\beta}{dt} \end{bmatrix} = \psi = \Phi \begin{bmatrix} \psi_R \\ \psi_L \end{bmatrix} \tag{7}$$

Φ is the gradient matrix.

The position error vector can be calculated as

$$\frac{dE}{dt} + KE = 0 \tag{8}$$

where $E_i = E_{i,0}\, f\left(exp(-Kt)\right)$

By using the above equation (8) the velocity of right and left wheel can be calculated as

$$\begin{bmatrix}\psi_R \\ \omega_L\end{bmatrix} = \phi^{-1}\psi = K\left(\alpha_\xi - \alpha\right) + \dot{\alpha_\xi} \tag{9}$$

The above presented mathematical models are for controlling the motion of the robot and finding the stability conditions. The next step is to design a model for the avoiding the obstacles. This model is defined below.

In order to navigate the robot autonomously a neural network based model is developed which helps in navigating the robot by providing the control on the motion and avoiding the obstacles. It takes the user defined inputs which are given as in terms of the co-ordinates. The architecture of model is shown in Figure 5.

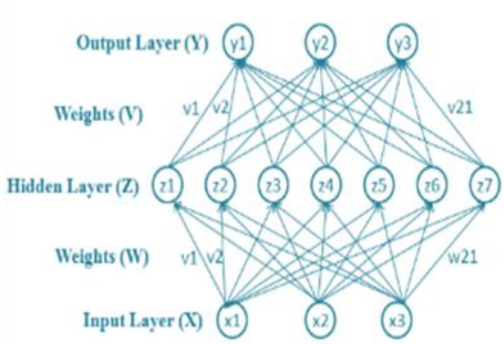


Figure 5. Architecture of Neural Networks

The above given architecture shows the neural network representation. The network consists of three nodes x1, x2, and x3. The input data is assigned to the robot is the initial co-ordinates and the travelling position co-ordinates. For each node or input the weight is calculated and passed through the hidden layer. The output layers also consist of three nodes y1, y2, and y3.

The neural network takes the input data and performs the training on the data which is defined below:

(i) Neural weight initialization.
(ii) Training of the samples
(iii) Calculation of hidden output layer data, hidden layer data based on the weights.
(iv) Find the error in weight calculation
(v) Minimization of error until network is well trained.

The error present in the network can be calculated as

$$\mathcal{E}(\alpha) = \frac{1}{2}\sum_{0}^{K\text{-}1}(E_K - \psi_k)^2 \tag{10}$$

Since the training process is performed for each sample one by one, weights also need to be updated at each step which is defined as

$$w(t_{iter} + 1) = w(t_{iter} + \varphi\sigma_i\psi_i) \tag{11}$$

φ is the learning rate

σ_i error gradient

An overall system works as follows (i) selection of neurons in the hidden layer and output layer and defining the learning rate. Then the next step is to perform training until the satisfactory performance value of training is not achieved. Figure 6 shows the overall architecture of our method.

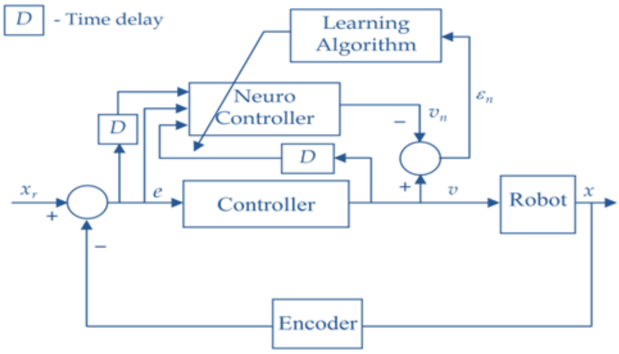


Figure 6. Neural Network used in our System.

IV. Results and Discussion

This section presents the simulation results of autonomous robot navigation using neural network. According to our proposed model we have considered Khepra-III mobile robot. Parameters which we have considered for the simulation of Khepra-III robot are tabulated below.

Table 2. Autonomous Robot Simulation Parameters

Parameter Constraint	**Constraint Value**
Diameter	55 mm
Height	30 mm
Empty weight	80 g
Speed	0.02 to 1.0 m/s
Autonomy	45 minutes moving
Processor	Motorola 68331 CPU @ 16 MHz
RAM	256 KB
EEPROM	512 KB
System	Running µKOS RTOS
2 DC brushed servo motors with incremental encoders	
8 infrared proximity and ambient light sensors (SFH900)	

Initially, robot movement area is designed. Keeping in mind the end goal to assess, the normal execution of our approach over different situations, we performed simulation of the neural systems for robot navigation for various environments. We can change the position of obstacle so we get other diverse situations. These situations were arbitrarily created .To locate a possible and right way after insertion of erasure of an impediment, we recreate the conduct of our self-learning so as to govern portable robot and applying the neural system standard. In our proposed approach robot is simulated for various environments. To reflect the robot conduct procured by learning in the investigated environment and in new unvisited situations. The robot responds in productive and an attractive way in these situations. As should be obvious the speculation and adjustment capacities of the framework are accomplished in terms of error minimization performance of the model as the number of iteration for a particular path increases, the error also minimized. The design of the situations changes by including different states of static impediments, in every circumstance the robot can explore effectively which can be seen from the various simulation scenarios. The proposed methodology can bargain a wide number of situations and provides for our robot the self-ruling choice of how to keep away from deterrents and how to go to the objective. More, the path planning method covers the situations structure also, the engender separations through free space from the source position. The outcomes which are given in Figure 5 are exceptionally agreeable to see the efficiency of proposed model for autonomous robot navigation using neural network. The next step is to define the neural network parameters like training scale factor, total number of neurons in hidden layer and noise for training and testing. Training scale factor is the ratio of training the neural network, neurons presents the in hidden layer and the noise to check the robustness of the navigation model.

Figure 5(a) presents the user interface for the robot navigation. This is the area where robot performs the movement based on the coordinates. First of all the input parameters are given then the training operation is done using neural network which is shown in Figure 5(b).

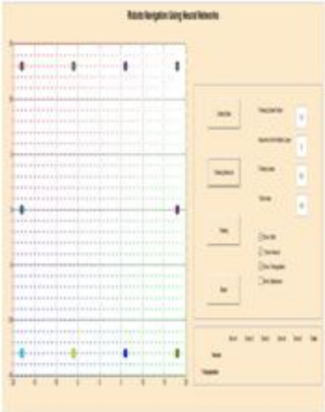


Figure 5(a) User Interface

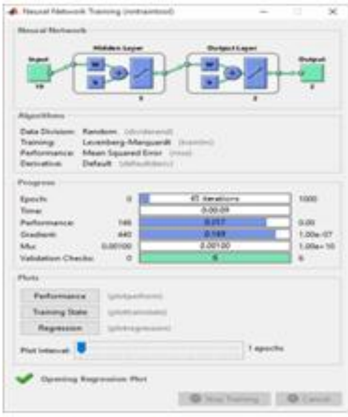


Figure 5(b) Neural Network Simulation

Training is an iterative process, so for each iterations the performance is shown Figure 5(c). Next step we perform the validation of the trained data which is given in Figure 5(d). In the Figure 5(e) the performance is shown with respect to the targeted data. Figure 5(f) presents the actual path of the robot in red dotted lines in next Figure 5(g) path followed by robot is shown in green lines and Figure 5(h) presents the path followed by SLAM algorithm [34]. It can be seen that the proposed model using neural network is more efficient compared to existing SLAM approach.

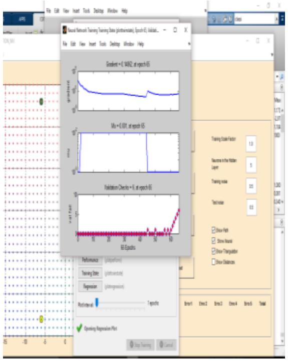


Figure 5(c) Neural Network Performance per epoch

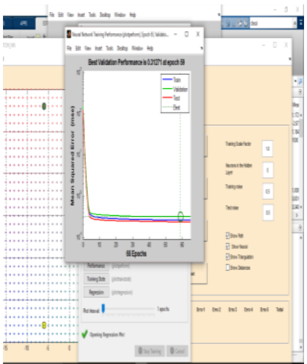


Figure 5(d) Training & Validation Performance

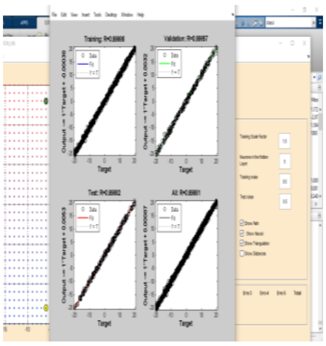


Figure 5(e) Neural Network Performance for Target

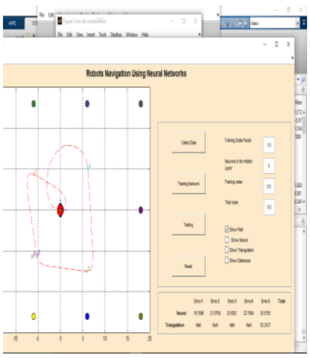


Figure 5(f) Robot Navigation for Original Path

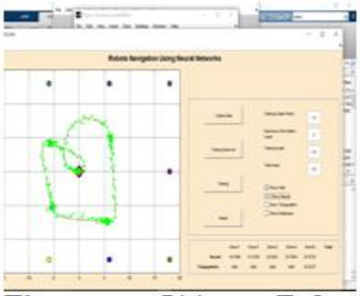


Figure 5(g) Robot Navigation using Neural Network

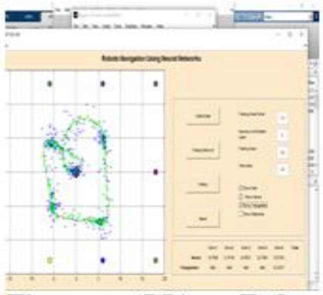


Figure 5(h) Robot navigation using Neural Network & SLAM

Figure 5. Simulation of Neural Network

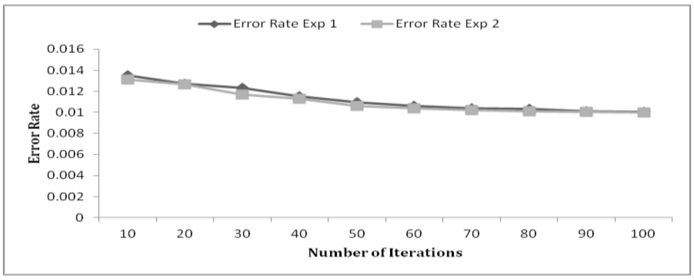


Figure 6. Iteration Vs Success Rate

The above given Figure shows the accuracy of the proposed system in the terms of success rate. As the number of training iteration is increases the success rate of achieving the target is also increases at the same time when success rate increases the error which is induced due to localization is also reduces. The localization error reduce is presented in Figure 7.

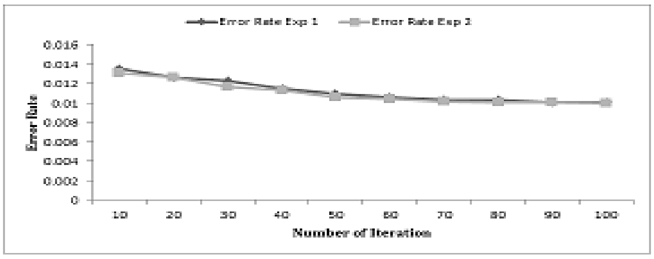


Figure 7. Localization Error Rate

Figure 8 and Figure 9 depict the regression and error plots for training ANN-A from our algorithm. As it can be seen from Figure 9, the mean squared error of the validation and test samples start to increase after epoch 13. Therefore to prevent the network from over fitting the training samples and to be able to generalize to new samples, training is halted at epoch 13. The regression plots in Figure 8 show the results of the networks outputs for the training patterns compared to the actual targets at step 13.

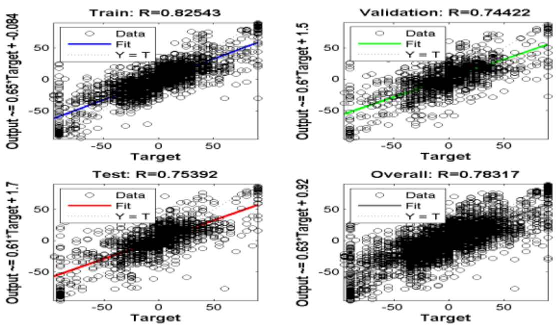


Figure 8. Regression Plots for Training ANN with 3000 Training Samples and 3000 Samples for Testing and Validation.

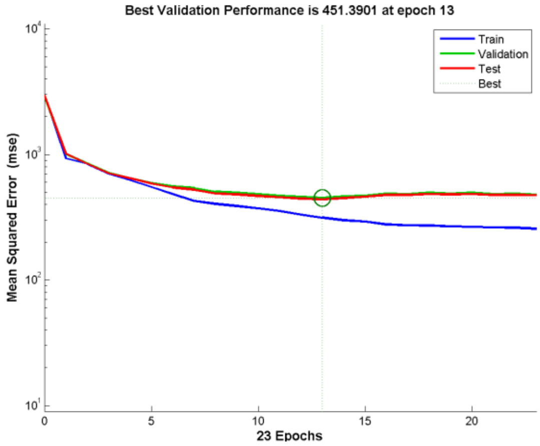


Figure 9. Performance Plot for Training ANN

Regression and performance plots of training results for keep-left and keep-right wall following networks are shown in Figure 10 through Figure 11. Based on the performance plots we can see that the networks have obtained the best validation performance for training ANN at epochs 16 and 12 respectively. The plots show very good results for the laser scanner patterns (training samples). Although the accuracy in comparison to the validation and testing patterns are a bit lower than the training samples, the final result is satisfying.

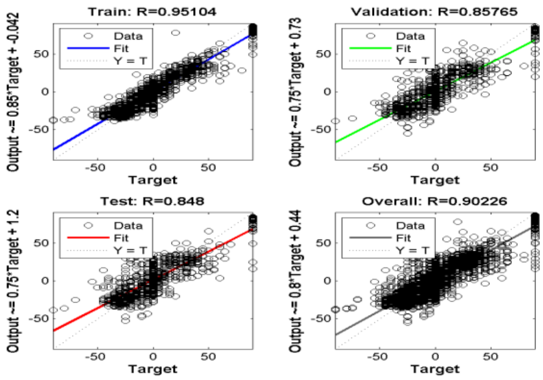


Figure 10. Regression Plots for Training ANN with 1500 Training Samples and 1500 Samples for Testing and Validation.

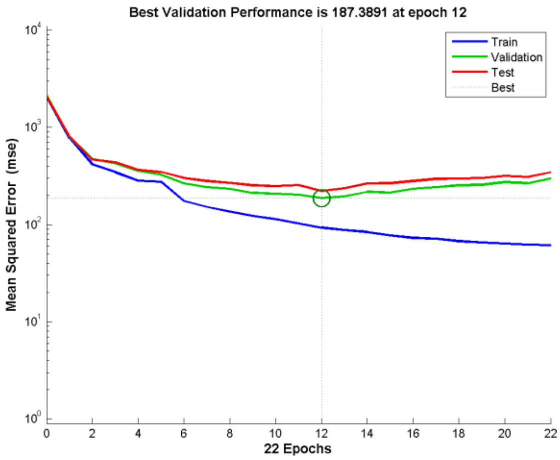


Figure 11. Performance Plot for Training ANN

IX. Conclusion

In this paper we have proposed a novel algorithm for autonomous robot navigation and control using neural network. Overall working of proposed system is divided in to two parts first is planning the path and the second is avoiding the obstacles. Path planning is done using kinematics mathematical model and avoiding the obstacle is performed by using the neural network controller. Neural network performs the training on the input samples and predicts the obstacles. Based on the training of the neural network, training error can be calculated. Based on the weights of the hidden layer and output layer, it performs the prediction to obstacle, if obstacle is present there on the path then by using our path planning model it finds the another path. Starting out from a start location and orientation in the grid, the mobile robot can autonomously head for destination Cells. On the way it determines its location in the grid using the principle of the neural networks. We demonstrated how we implemented the underlying algorithm in software. Target location situations are associated with favorable actions in an obstacle-free environment explained in detail in this paper. Simulation is done using MATLAB tool and the results shows that the proposed model of the robot navigation is an efficient and robust to the environment and environment noise.

References

[1] Li P., Huang X., Wang M., "A New Hybrid Method for Mobile Robot Dynamic Local Path Planning in Unknown Environment", Journal of Computers, North America, Vol. No. 5, pp. 773-781, 2010.

[2] Joshi, M.M., Zaveri, M.A., "Neuro-fuzzy based autonomous mobile robot navigation system," in 11th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 384-389, 7-10 Dec. 2010

[3] Mohamed, A.Z., Sang Heon Lee, Hung Yao Hsu, "Autonomous mobile robot system concept based On PSO path planner and vSLAM," in IEEE International Conference on Computer Science and Automation Engineering (CSAE), Vol. No. 4, pp. 92-97, 10-12 June 2011

[4] Sariff, N., Buniyamin, N., "An Overview of Autonomous Mobile Robot Path Planning Algorithms," in Research and Development, 2006. 4th Student Conference on Research and Development, pp. 183-188, 27-28 June 2006

[5] Dei-Jeung Huh, Jong-Hun Park, Uk-Youl Huh, Hak-Il Kim, "Path planning and navigation for autonomous mobile robot," in IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the] , Vol. No. 2, pp. 1538-1542 5-8 Nov. 2002

[6] Qingxiang Wu, Bell, D.A., Khokhar, R.H., Guilin Qi, Zhihui Luo, "Autonomous Robot Control Using Evidential Reasoning," in Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Vol. No. 4, pp. 550-554, 24-27 Aug. 2007

[7] R. Frizera Vassallo, H.J. Schneebe li, J. Santos-Victor, "Visual navigation: combining visual servoing and appearance based methods", International Symposium on Intelligent Robotic Systems, SIRS'98, Edinburgh, Scotland, July 1998.

[8] Yasuda, G., "Behavior-based autonomous cooperative control of intelligent mobile robot systems with embedded Petri nets," in 7th Joint International Conference on 15th International Symposium on Soft Computing and Intelligent Systems (SCIS), and Advanced Intelligent Systems (ISIS), pp. 1085-1090, 3-6 Dec. 2014

[9] Ziemke, T., "Remembering How to Behave: Recurrent Neural Networks for Adaptive Robot Behavior". In: Recurrent Neural Networks: Design and Applications, CRC Press 2000. ISBN 0849371813. pp. 355–390

[10] Jasmin Velagic, Nedim Osmic and Bakir Lacevic, "Design of Neural Network Mobile Robot Motion Controller" INTECH open science, 2010

[11] Pinto, A.B., Barbosa, R.S., Silva, M.F., "Neural control of an autonomous robot," in IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp .339-344, 14-15 May 2014

[12] Hasan, K.M., Abdullah-Al-Nahid, Reza, K.J., "Path planning algorithm development for autonomous vacuum cleaner robots," in International Conference on Informatics, Electronics & Vision (ICIEV), 2014, pp. 1-6, 23-24 May 2014

[13] Maykol Pinto, A., Costa, P.G., Moreira, A.P., "An architecture for visual motion perception of a surveillance-based autonomous robot," in IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 205-211, 14-15 May 2014

[14] Engedy, I., Horvath, G., Artificial Neural Network Based Mobile Robot Navigation. Hungary, International Symposium on Intelligent Signal Processing, 2009, 241 – 246

[15] A. Saffiotti, "The Uses of Fuzzy Logic in Autonomous Robot Navigation: a Catalogue Raisonne, Soft Computing", 1997, pp. 180-197.

[16] C. Thomaz, M. Pacheco, M. Vellasco, "Mobile Robot Path Planning Using Genetic Algorithms", (Lecture Notes in Computer Science, Springer Book Series, Vol. No. 1606, pp. 671-679).

[17] Kuo-Ho Su, Feng-Li Lian, Chan-Yun Yang, "Navigation design with SVM path planning and fuzzy-based path tracking for wheeled agent," in International Conference on Fuzzy Theory and it's Applications (iFUZZY), pp. 273-278, 16-18, Nov. 2012

[18] Goyal, J.K., Nagla, K.S., "A new approach of path planning for mobile robots," in International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 863-867, 24-27 Sept. 2014

[19] Yunfei Zhang, de Silva, C.W., Dijia Su, Youtai Xue, "Autonomous robot navigation with self-learning for collision avoidance with randomly moving obstacles," in 9th International Conference on Computer Science & Education (ICCSE), pp. 117-122, 22-24 Aug. 2014

[20] Becerra, H. M., G. Nicol'as, L., SagOOu'es, C., "A SlidingMode-Control law for mobile robots based on epipolar Visual servoing From Three Views," IEEE Transactions on Robotics, 2011, Vol. No. 27, pp. 175-183.

[21] Lapierre, L., Zapata, R. and Lepinay, P., "Combined path following and obstacle avoidance control of a unicycle-type robot," International Journal of Robotics and Research, 2007, vo1.26, No..4, pp. 361-375.

[22] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, D. Vallejo, "OBPRM: An obstacle-based PRM for 3D workspaces", In Proceedings of. third International Workshop on the Algorithmic Foundations of Robotics, 1998.

[23] E. G. Gilbert and S. M. Hong, "A new algorithm for detecting the collision of moving objects", In IEEE International Conference on Robotics and Automation, Vol. No. 1, pages 8–14, Scottsdale, AZ, USA, May 1989.

[24] Iijima, J., Yuta, S., and Kanayama, Y., "Elementary Functions of a Self-Contained Robot "YAMABICO 3.1", Proceedings . of the 11th Int. Symp. on Industrial Robots, Tokyo, 1983, pp. 211-218.

[25] M. Habib, "Real Time Mapping and Dynamic Navigation for Mobile Robots", International Journal of Advanced Robotic Systems, , ISSN 1729-8806, pp. 323-338, Vol. No. 4, 2007.

[26J. Borenstein, H. R. Everett, and L. Feng., "Navigating Mobile Robots: Systems and Techniques", A. K. Peters, Ltd., 1996

[27] J. A. Castellanos and J. D. Tard´os, "Mobile Robot Localization and Map Building: A Multisensor Fusion Approach", Kluwer Academic Publishers, 1999.

[28] T. Lozano-P´erez, J. L. Jones, E. Mazer, P. O´Donnel, W. E. L. Grimson, P. Tournassoud, and A. Lanusse. Handey: "A robot system that recognizes, plans, and manipulates", In IEEE International Conference on Robotics and Automation, Vol. No. 4, pp. 843–849, Mar. 1987.

[29] L. Jones, Joseph and T. Lozano-P´erez., "Planning two-fingered grasps for pick-and-place operations on polyhedral", In IEEE International Conference on Robotics and Automation, Vol. No. 1, pages 683–688, Cincinnati, OH, May 1990.

[30] M. Suppa, P. Wang, K. Gupta, and G. Hirzinger, "C-space exploration using noisy sensor models", In IEEE International Conference on Robotics and Automation, Vol. No. 5, pp. 4777–4782, New Orleans, LA, USA, Apr. 2004.

[31] T. Fujii, Y. Arai, H. Asama, and I. Endo, "Multilayered reinforcement learning for complicated collision avoidance problems", in IEEE International Conference on Robotics and Automation, 1998, pp. 2186– 2191.

[32] K. Ishii, S. Nishida, K. Watanabe, and T. Ura, "A collision avoidance system based on self-organizing map and its application to an underwater vehicle," 7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002, pp. 602-607.

[33] S. Grossberg and D.S. Levine, "Neural dynamics of attentionally modulated Pavlovian conditioning: blocking, interstimulus interval, and secondary reinforcement," Applied Optics, Vol. No. 26, 1987, pp. 5015-5030.

[34] T. Bailey, J. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2006.