

An ANN Based NARX GPS/DR System for Mobile Robot Positioning and Obstacle Avoidance

Ramazan Akkaya^{1,*}, Omer Aydogdu¹, Suleyman Canan²

¹Department of Electrical and Electronics Engineering, Selcuk University, Konya, Turkey

²Research Dept. of Elfatek Electronics Corporation, Konya, Turkey

*Corresponding author: akkaya@selcuk.edu.tr

Received April 24, 2013; Revised May 15, 2013; Accepted May 16, 2013

Abstract Conventional sensor integration and navigation methods are based on the Kalman filter algorithm. Kalman filter needs a pre-defined model of the dynamic system. In most of the case non-linear system modeling might be a changing computational load. Artificial Neural Network (ANN) computing is a very powerful tool for solving non-linear problems involving mapping input and output relation without any prior knowledge of the system and the environment involved. This study has investigated Global Positioning System (GPS) and Dead Reckoning (DR) sensor fusion approach using ANN Nonlinear Autoregressive with external input (NARX) model. The ANN accepts navigation sensor data and is trained throughout a pre-design training track for gathering training data set which is used to predict mobile robot position where GPS signal is lost. In addition, a simple obstacle avoidance algorithm has been added to the system because the mobile robot can find its own trajectory again by circulates around the obstacle. The experimental results for different test data examples demonstrate that the proposed ANN NARX sensor fusion model can be used for reliable position and heading estimation of the mobile robot.

Keywords: *wheeled mobile robot, kalman filter, ANN NARX model, dead reckoning, obstacle avoidance*

1. Introduction

Mobile robots will have increasing significance in our daily life. They make it possible for law enforcement, domestic uses where repeated tasks are needed, exploration, hazardous material handling, and military field and surveillance applications. A mobile robot may be subdivided in different parts: navigation, sensors, computing power, hardware, software, mechanical, and communication [1,2]. All these parts affect in a way the navigation and capabilities of the mobile robot.

Two questions reside in a successful mobile robot design:

- Going to a desired target following a near optimal path between its current positions.
- Determining the mobile robot position in some absolute coordinate system.

This work is mainly focused on the second problem. For this purpose the navigation structure stands around a Global Positioning System (GPS) as absolute positioning system, and Dead Reckoning (DR) relative positioning sensors integration concept.

Complimentary aspect of both navigation type leads to a continuous positioning in urban areas, foliage and harsh environment where the number of GPS satellite view may drop under four satellites or completely loss of GPS signal. GPS is a very good absolute position system but has the vulnerability to environmental obstruction causing signal loss and multi-path degenerating GPS positioning accuracy [3,4]. DR navigation system generates fairly

good relative positioning in short-term navigation. However DR has the disadvantage of accumulating integration errors. Relying only on one type of navigation system would not be a good choice, where interruption of position data may cause fatal problems. Multi-sensor integration philosophy is to fuse all the data available from sensors in order to obtain an optimal navigation solution [5]. Blending GPS/DR sensor will generate viable and continuous positioning data even when GPS signal is blocked for short time. Traditionally, a Kalman Filter is used for sensor fusion, which contains errors to obtain optimal estimate [6,7]. The drawback of the Kalman Filter is that the process dynamic model should be known. The heavy computational work and matrix operation in the Kalman Filter algorithm are other considerations to be taken into account where low cost system design is envisaged. In contrast ANN has not the obligation to know the system model or the environment. The only requirement for ANN is the training data set. ANN learns through examples presented to its inputs and comparing with the wanted outputs [8]. The learning process continues on defined iteration number or constraints such least square errors limits.

Also, obstacle detection and avoidance in a real world environment - that appears so easy to humans is a rather difficult task for autonomous mobile robots and is still a well-researched topic in robotics [9]. The autonomous guided vehicle is equipped with two-level sensor systems, ultrasonic transducers and odometry, which allow it to estimate and correct its global position continually [10]. Many research robots also have sonar type sensors, emitting a signal and then listening for echoes that

indicate objects. In mobile robot application, ultrasonic sensors are used widely due to the recognition of objects with the image of reflection [11,12,13].

The aim of this paper focuses on a multi-sensor integration method using ANN NARX architecture, a data collection strategy for the neural network training data set and the evaluation of the proposed neural network model. Also, a simple obstacle avoidance algorithm is improved for mobile robot so that the mobile robot can find its own trajectory easily by circulates left around the obstacle.

2. The Wheeled Mobile Robot Platform Description

A car like mobile robot [14] was built in order to interface two encoders for each of the back wheels, the GPS antenna and fiber optic gyroscope (FOG) was placed above the rear wheel centre axis of the mobile robot (see Figure 1(a)). The mobile robot is equipped with one low cost L1 single frequency GPS, which generate position data every second, a FOG giving angle rate measurements at 10Hz, two URM04 V2.0 ultrasonic sensors to detection of any obstacle, and two odometer sensors with 500 hundred slots each. On board, a PC104 computer is used to control the mobile robot and gather the sensors data. The mobile robot has also a remote control system for steering and start/stop functions.

The wheels diameter of the mobile robot is 29.85cm; the rear axis has a distance of 59cm and front/back wheel distance of 67cm. Thus the theoretical resolution of the odometer sensor is about 0.1875cm/slot (see Figure 1(b), 1(c)).

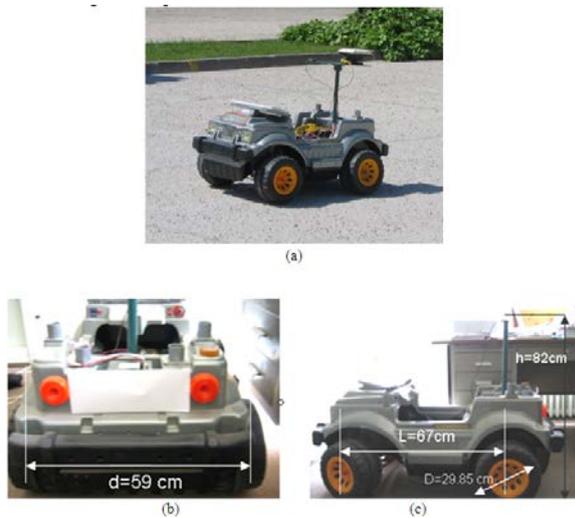


Figure 1. (a) Mobile robot in an experimental work (b) back view (c) lateral view

The hardware structure given in Figure 2 consist of a PC104 single board computer as a core computing engine with a 300 MHz CPU, 2 RS232 serial port, 1 RS485 serial port and 64 MB RAM memory, and 32MB of flash disk for DOS operating system and data storage. A dedicated electronic hardware circuitry interfaced to the PC104 ISA bus is designed to interface odometer sensors and power drive electronics. The PC104 may control the wheel driver motor and the steer motor of the mobile robot. The FOG angle rate data and GPS position data is collected via the

RS232 serial Ports. The PC104 saves the collected sensors raw data on the flash disk and time stamps for each sampled data. The whole sensors are synchronized with the FOG data which set a bit every 1/10 second in the outputted data stream for time synchronization with other sensors. A control and data logging software was written with ANSI C language. The odometer sensors pulses are counted by two 24 bit digital counters.

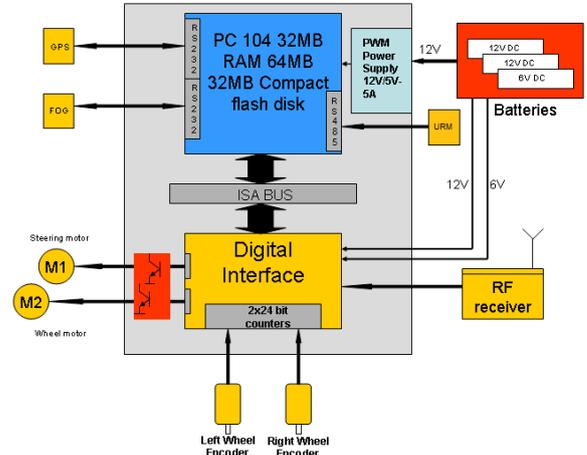


Figure 2. Sensors data gathering hardware structure

2.1. The Mobile Robot Kinematic Model

Consider the coordinate system shown in Figure 3. The mobile robot's planar coordinate system (x, y) and heading (ϕ) describe the position of the mobile robot. The encoders count and FOG measurements are treated as inputs to the system.

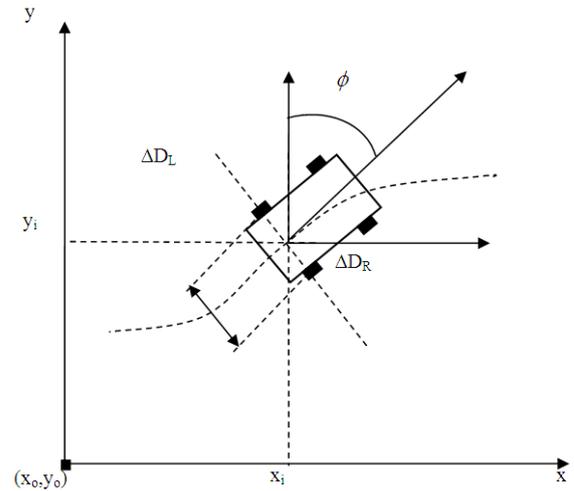


Figure 3. The mobile robot coordinates

The mobile robot is modeled by the following nonlinear kinematic equations representing the mid-axis (x, y) and (ϕ) the mobile robot orientation in the global frame,

$$\begin{aligned} x(k+1) &= x(k) + \Delta D(k) \cdot \cos(\phi(k) + \frac{\Delta \phi(k)}{2}) \\ y(k+1) &= y(k) + \Delta D(k) \cdot \sin(\phi(k) + \frac{\Delta \phi(k)}{2}) \\ \phi(k+1) &= \phi(k) + \Delta \phi(k) \end{aligned} \quad (1)$$

Where $\Delta D(k)$ is the average of $\Delta D_R(k)$, $\Delta D_L(k)$, the right and left wheel traveled distance respectively. The kinematic model equations are used to calculate the mobile robot position in pure dead-reckoning and compare with the proposed neural network positioning method.

3. The Mobile Robot Navigation System Design

Autonomous navigation in unknown environments is a key capability of a mobile robot [15]. There are many different methods in designing a navigation system for a mobile robot. The commonly used algorithm is the extended Kalman filter. Wheeled mobile robots are nonlinear dynamical systems, their kinematics involves geometrical nonlinearities. Thus, the identification of these systems requires nonlinear models. Neural Network has been widely used in control systems, system identification signal filtering and classification applications. Especially ANN system identification is very well suited to solve nonlinear problems. Any prior knowledge of the system is not needed for the ANN model. ANN can be defined as universal model. ANNs are powerful tools for solving nonlinear problems, involving input data to output data mapping. The learning algorithm used in a variety of ANN is the backpropagation algorithm. The backpropagation learning rule is central to much current ANN work on learning [16]. The backpropagation algorithm is a gradient descent. Meaning that, the network error is back propagated from the output through the entire hidden layer to the input layer. The network error is used to adjust the weights associated with the connection and neurons by applying a generalized Delta rule [17]. The weights are adjusted till the input and output relation is established under a define condition.

3.1. The NARX Neural Network Model for Mobile Robot

The artificial neural network was designed to imitate the very basic operation of a biological neuron. Essentially a set of inputs are applied each representing the output of another neuron, where multi layer ANN is used. Each input is multiplied by a corresponding weight and all the weighted inputs are then summed to determine the activation level of the neuron [18], Figure 4 depicts a model implementation of the above functional description. Equation (2) describes the mathematical relationship between the inputs and output of the neuron.

$$\begin{aligned} Y_{net} &= w_1 z_1 + w_2 z_2 + \dots + w_i z_i \\ &= \sum_{j=1}^i w_j z_j \\ Y_{out} &= f(Y_{net}) \end{aligned} \quad (2)$$

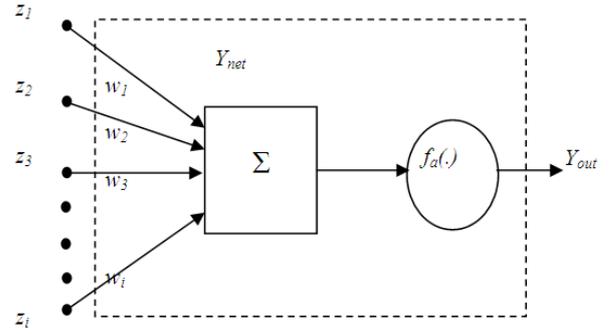


Figure 4. Basic artificial neuron model

An important functional block of an ANN is the activation function. Typically non-linear activation function $fa(.)$ in the form of a hyperbolic tangent is used in the hidden layer of a multi-layer ANN for input and output data mapping of non-linear systems, and linear activation functions in the inputs and outputs of the ANN.

The problem of identification of nonlinear dynamic systems using ANN NARX model is addressed in this paper. The ANN NARX model is extended for MIMO (multiple input multiple output) systems. Let's define the physical input/output of a time-discrete dynamic system at time t as $u(t)$ and $y(t)$, respectively.

In Equation (3), $g(.)$ is the regression function performed by the ANN θ weightings of the inter-neuron connections, ψ is the past inputs and outputs data.

$$\begin{aligned} \hat{y}(t) &= g[\psi(t, \theta), \theta] \\ \varphi(t) &= [u(t-1) \dots u(t-m), y(t-1) \dots y(t-n)]^T \end{aligned} \quad (3)$$

The presented ANN NARX model in Figure 5 has no feedback from its predicted outputs; consequently the ANN NARX is a stable structure [19]. This scheme is the preferred choice when the system noise is insignificant. The input/output training data set was gathered from the reference training trajectory given in Figure 2 and the data is presented to the ANN NARX model in Figure 5. A free MatLab toolbox named NNSYSID developed by Norgaard and his friends [20] is used in this work.

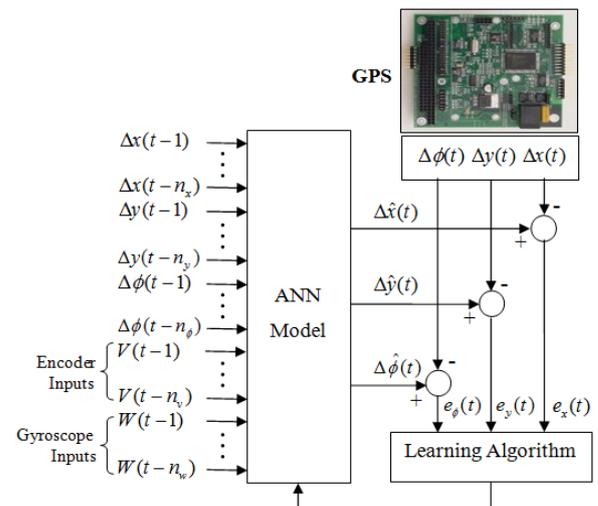


Figure 5. ANN NARX model

3.2. GPS/DR Integration Using ANN NARX Method

The ANN NARX navigation method will be applied to GPS/DR integration problem. This sensor integration method differs from the method given in [21,22] by the input structure. Past input/output sensors data are not used in this method [21,22]. In addition this method uses a defined set of training data that speed up the learning process of the ANN. To apply this method firstly a reference training trajectory was chosen, and then the structure of the ANN is chosen.

The ANN should be trained off-line first in order to solve the non-linear input/output relationship. When the GPS data is not available the ANN will be in prediction mode with DR sensors that are always available using the existing previous trained weight values. Otherwise the GPS data will be used to reinitialize the mobile robot position. In addition, the outputs $[\Delta x(t), \Delta y(t), \Delta \phi(t)]$ are obtained by differentiating previous GPS outputs with the current GPS data, thus making ANN training more efficient and saving extra time.

Eight past mean velocity values measured from the optical encoders $[V(t-1), \dots, V(t-8)]$ eight gyro heading values $[\omega(t-1), \dots, \omega(t-8)]$, and no past output values are presented to the network input as shown in Figure 5. In this special case the model is transformed to a NFIR (Nonlinear Finite Impulse Response) model. The network outputs the predicted difference between epochs, $[\Delta \hat{x}(t), \Delta \hat{y}(t), \Delta \hat{\phi}(t)]$ for the North component, the East component, and for GPS heading respectively. Thus eight measured data values are needed in order the network give response when prediction mode is started. The mobile robot predicted coordinates are calculated by integrating

the network outputs to the initial values as given in Equation (4).

$$\begin{aligned}\hat{x}(t+1) &= \hat{x}(t) + \Delta \hat{x}(t) \\ \hat{y}(t+1) &= \hat{y}(t) + \Delta \hat{y}(t) \\ \hat{\phi}(t+1) &= \hat{\phi}(t) + \Delta \hat{\phi}(t)\end{aligned}\quad (4)$$

3.3. Generating Training Data for the ANN

To train a neural network to estimate (x, y, ϕ) coordinate and heading from input feature vector. It is necessary to collect enough input for ANN data to cover almost every possible range of the input space domain. The training data should be evenly distributed throughout the environment. A speed range of 0 to 0.9m/sec has been recorded after several trial run of the mobile robot. The mobile robot speed slows down when tuning a sharp curvature to the right or left and on ascending hills, speeds up when descending hills, stay constant around 0.6m/s on smooth straight tracks. The mobile robot driving motor is not speed controllable. However the driving motor adapts its speed against the load applied from external frictions and cited causes above, when navigating. The mobile robot heading value ranges between $(-\pi, +\pi)$ rad. Every sample between each second was integrated, therefore decreasing the data dimension of gyro samples and synchronizing with the GPS data rate. From these specifications of the mobile robot, a training data generation scheme has been determined. It is found that if the mobile robot follows concentric ellipses track as shown in Figure 6 would be enough to train the paper [23].

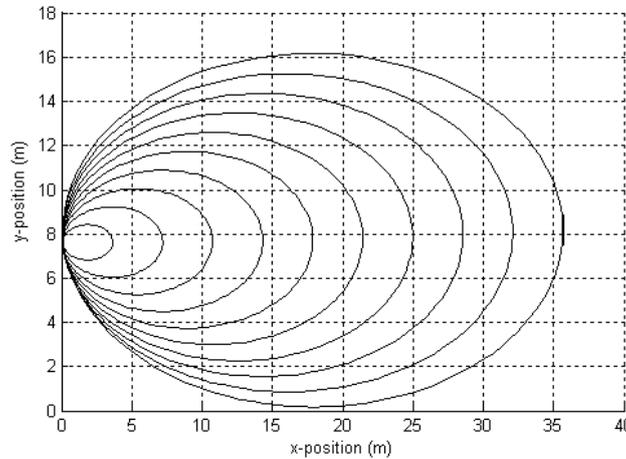


Figure 6. Theoretical reference training trajectory

Table 1. RMS position error versus ANN iteration number

Iteration number	RMS position error(m)
5	2.104
10	0.962
50	0.920
100	0.909
200	0.904
250	0.910
300	0.909
400	0.908
500	0.908
1000	0.907
1500	0.907

Several training iteration have been tested and the different iteration numbers results are shown in Table 1.

We may see that an iteration number over 300 doesn't make much improvement in the overall positioning RMS error. In this situation a 300 learning iteration number was chosen to train the ANN NARX model. By trial and error 20 hidden neurons with tangent hyperbolic activation function was chosen and linear activation function was used for the input and output neurons.

3.4. The Mobile Robot Obstacle Avoidance Algorithm

In designed mobile robot system, the robot trajectory is determined via GPS and learning algorithm, and then it is desired that this trajectory is followed by mobile robot

exactly. If mobile robot encounters with an obstacle on their path, the robot rotates to the left automatically and goes on sufficiently. Then, the mobile robot turns to the right and passes to the obstacle. Later, the mobile robot turns to the right again and finds their trajectory to continue own path [24]. The algorithm for obstacle avoidance that developed in this research is shown in below.

- Start (If there is any obstacle),
- Determine obstacle distance and width,
- Turn left and proceed throughout the obstacle width,
- Turn right and determine obstacle distance and length,
- Proceed throughout the obstacle length and turn right again,
- Proceed to find the trajectory and follow the path.

4. Field Results

The experimental reference training trajectory has been generated; the true GPS trajectory is shown in Figure 7 with solid line. The reference track generated is tried to keep as similar as to the theoretical track, however due to area limitation in northing direction in the working field; the experimental reference track was clipped mostly in that direction. The starting point is shown as a star marker in Figure 7 and the mobile robot movement is in clockwise direction. The resulting ANN NARX predicted trajectory is shown in dashed line in Figure 7.

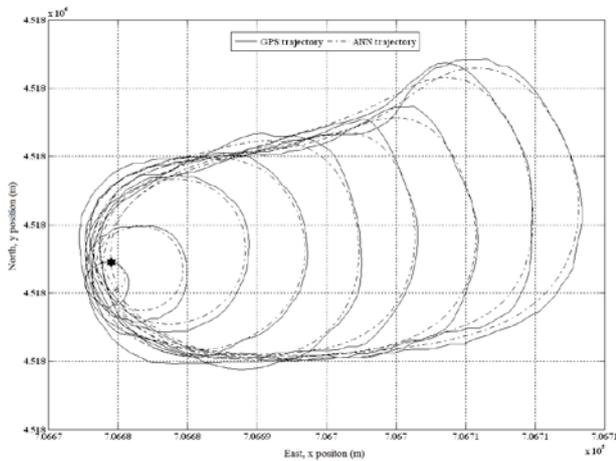


Figure 7. Experimental reference with GPS and ANN NARX predicted trajectory

In Figure 8 we may observe the RMS position errors with a small trend along the predicted reference trajectory. The errors tend to increase as navigation time increase, this behavior is caused by the integration of the ANN outputs errors. An overall of 0.86m positioning RMS error is calculated where 5m positioning RMS errors obtained with stand alone DR navigation with gyro and optical encoders.

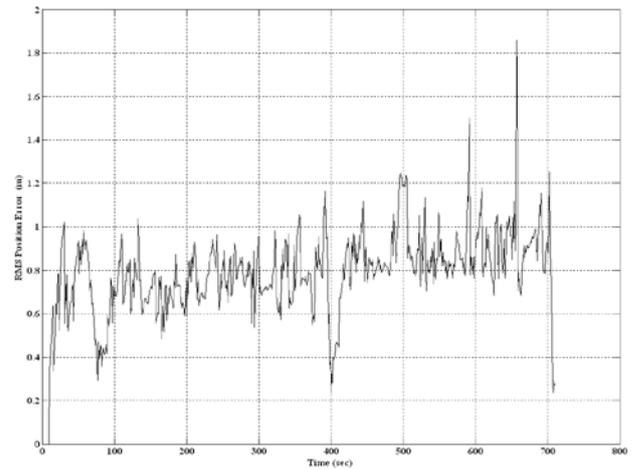


Figure 8. RMS position errors along the predicted reference trajectory

In Figure 9, the East and North errors are depicted, in both directions the errors remain almost below ± 1 meter which is quite acceptable in the proposed ANN NARX model.

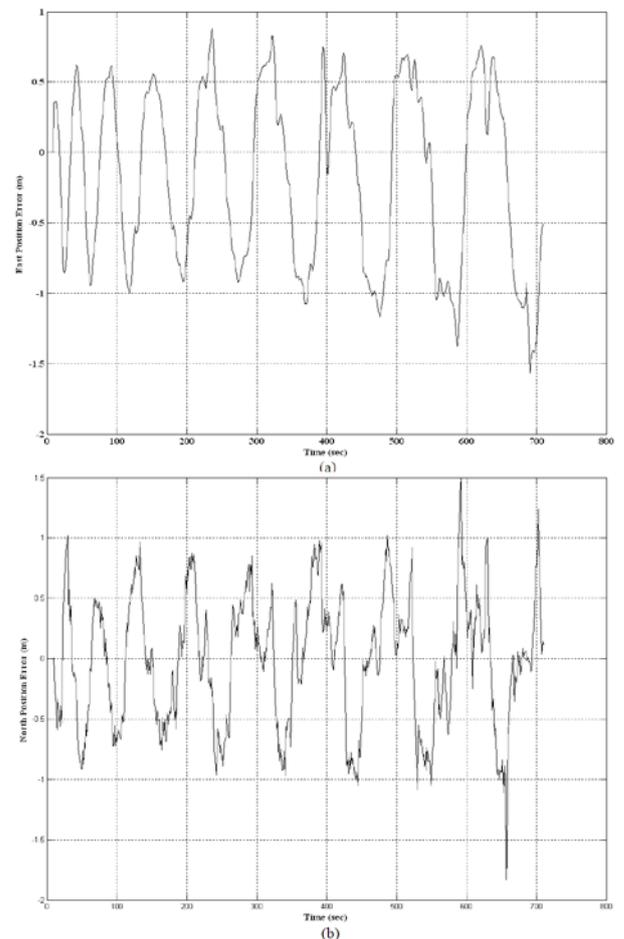


Figure 9. East (a) and north (b) errors along the predicted reference trajectory

A different trajectory has been followed in order to test the generalization capabilities of previously trained ANN NARX model. Figure 10 shows the true generated GPS and predicted trajectory. The total navigation time in the test trajectory is 375 seconds.

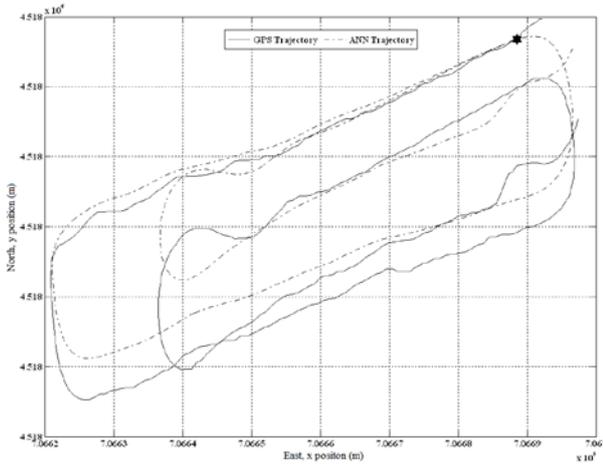


Figure 10. GPS and ANN NARX predicted test trajectory

The East and North errors graph are shown in Figure 11. The errors in Easting direction swing between ± 1 meters, but Northing errors increases with time. This is caused by sharp turns and integration o errors in North direction. It can be seen that the North errors contribute in the positioning RMS errors shown in Figure 12. ANN RMS error of 3.1m along the test trajectory is calculated only in prediction mode. As previously stated the ANN NARX model will predict the mobile robot position where GPS signal is lost.

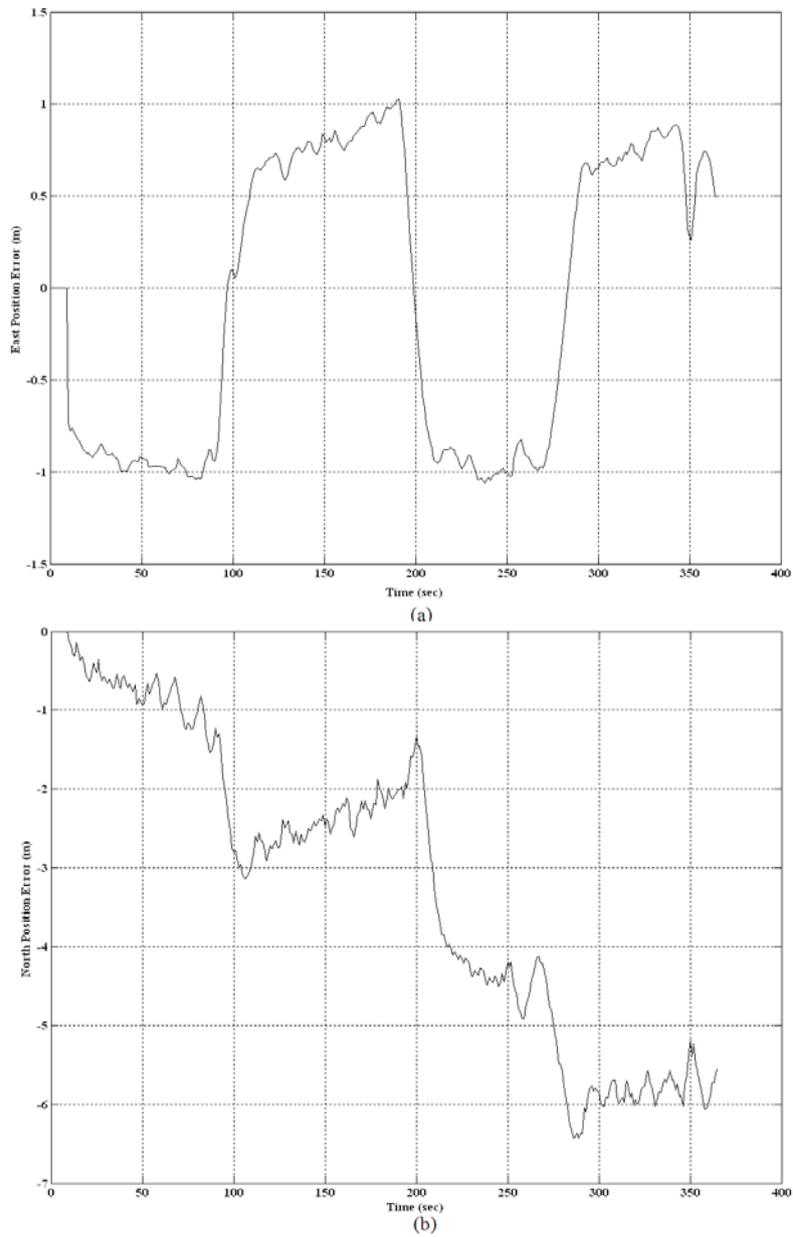


Figure 11. East (a) and north (b) errors in the predicted test trajectory

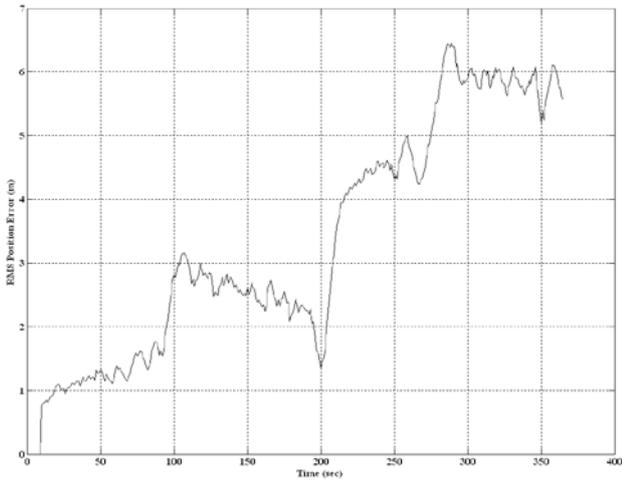


Figure 12. RMS position errors along the predicted test trajectory

Using an ANN NARX model trained with the reference trajectory seen in Figure 10 and resetting the mobile robot position when GPS position data is available. This configuration structure is shown in Figure 13.

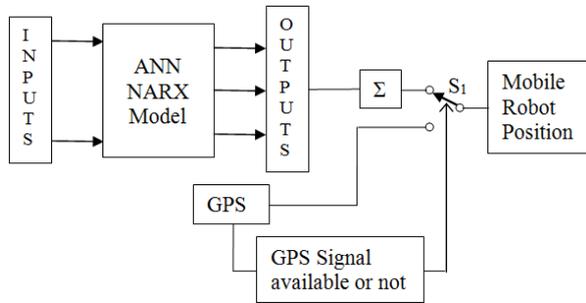


Figure 13. ANN predictor and GPS system

In this scheme, when GPS signal is available, the S1 switch select the GPS and reinitialize the current predicted mobile robot position, otherwise the ANN NARX model start running in prediction mode. The GPS availability time instant is determined randomly in order to simulate real GPS lock of loss and is given with the array defined below;

GPS_Availability_Times=(1,16,23,31,35,36,43,97,112, 122,135,137,157,165,168,181,192,197,200,213,214,217,2 22,227,229,239,243,247,262,269,288,290,291,294,301,30 4,307,331,342,350,358,366,372) second.

The simulation was conducted with the trained ANN NARX model and the GPS aided configuration as given in Figure 13. The predicted ANN NARX trajectory where the time the GPS positions are available is shown in Figure 14 with square markers along the test trajectory.

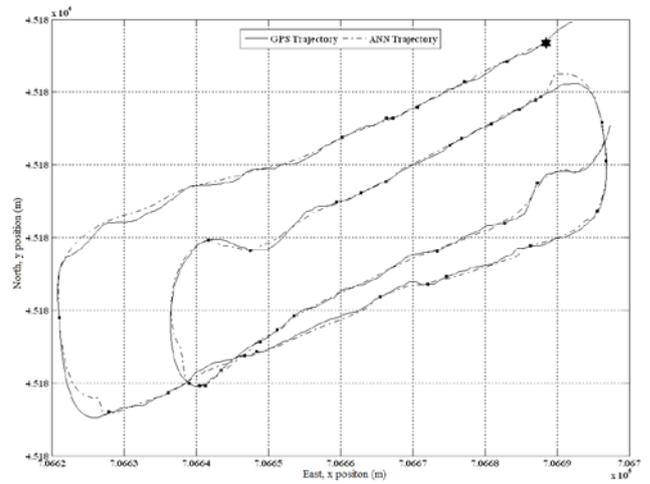


Figure 14. GPS and GPS aided ANN NARX predicted test trajectory

The RMS error is below 2.5m as shown in Figure 15. And in Figure 16 the errors in both directions decrease drastically with the aid of GPS.

Position and heading error comparison between ANN NARX and Kalman Filter model is given in Table 2. According to the results in Table, the RMS error drop below 1m with the GPS aid, and the trend given by MAE (Mean Average Error) decreases in both north and east directions.

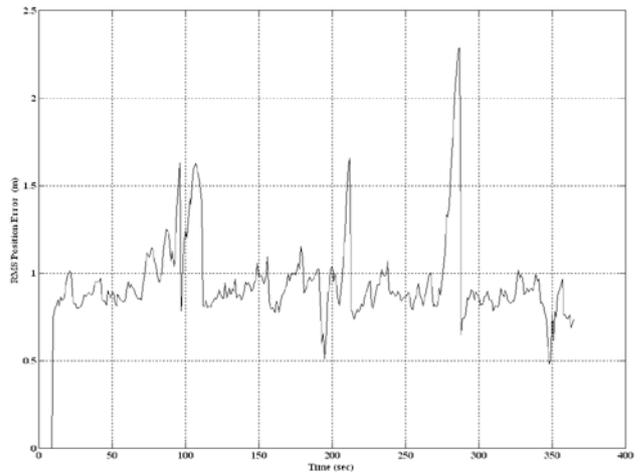


Figure 15. GPS aided RMS position errors along the predicted test trajectory

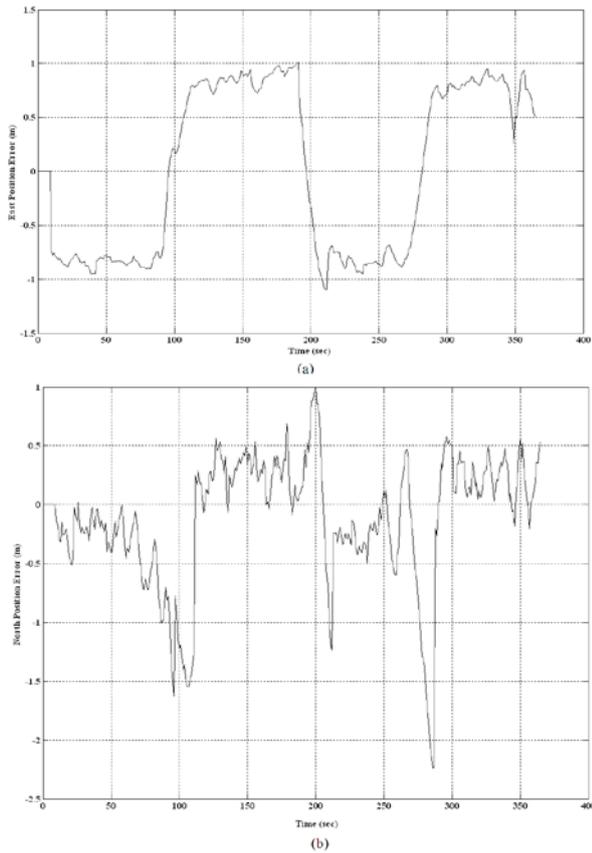


Figure 16. East (a) and north (b) errors in the predicted test trajectory

Table 2. Position and heading error comparison between ANN NARX and Kalman Filter model

Test Trajectory	ANN NARX Model
RMS position error without GPS signal (prediction)	1.80m
RMS position error with partial GPS signal (GPS integrated prediction mode)	0.960444m
MAE of East position with no GPS signal (prediction mode)	0.769903m
MAE of East position with partial GPS signal (GPS integrated prediction mode)	0.74m
MAE of North position with no GPS signal (prediction mode)	3.06m
MAE of North position with partial GPS signal (GPS integrated prediction mode)	0.40m

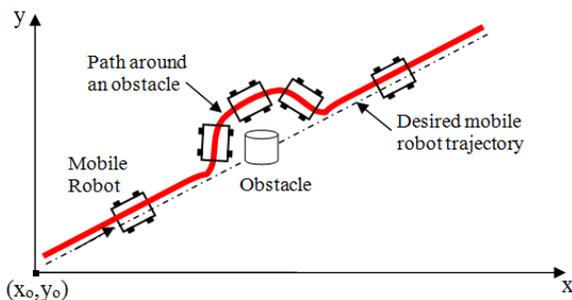


Figure 17. Desired mobile robot trajectory and path around an obstacle

5. Conclusion

The neural network is used here to model the mobile robot kinematics and this paper has shown that neural network can be used to integrate different navigation

sensors. The performance of this structure has been investigated with and without GPS aid.

The results given in Table 2 is examined, the RMS error drop below 1m with the GPS aid, and the trend given by MAE (Mean Average Error) decreases in both north and east directions. It has been shown that predicted position errors are acceptable and continuous position prediction process is successfully achieved from graphical plots and error analysis.

As a result, navigation using ANN-NARX sensor integration may be used as an alternative to the Kalman filter. It may be concluded that the ANN NARX structure can be used in alternative to conventional navigation methods. In addition, through obstacle avoidance algorithm is added to the system, the mobile robot can find its own trajectory by circulates left around the obstacle successfully.

Acknowledgement

This work was supported by Scientific Research Projects (BAP) Coordinating Office of Selcuk University.

References

- [1] Takafumi S. and Masaki T., Obstacle Avoidance for Autonomous Mobile Robots Based on Position Prediction Using Fuzzy Inference in Numerical Analysis - Theory and Application, Edited by Jan Awrejcewicz, Intech, Croatia, 2011, 577-588.
- [2] Canan, S., GPS Aided Neural Network Navigation System, PhD Thesis, Selcuk University, Turkey, 2006.
- [3] Mintsis, G., Basbas, S., Papaioannou, P., Taxiltaris, C. and Tziavos, I.N., "Applications of GPS technology in the land transportation system", European Journal of Operational Research, 152 (2), 399-409. Jan.2004.
- [4] Noureldin, A., El-Shafia, A. and Tahab, M. R., "Optimizing neuro-fuzzy modules for data fusion of vehicular navigation systems using temporal cross-validation", Engineering Applications of Artificial Intelligence, 20 (1), 49-61. Feb.2007.
- [5] Ashkenazi A., Moore, T., Dumville, M., Lowe, D. and Tsakiri, M. "An Artificial Intelligent Highway System", in Proceeding of the 8th International Technical Meeting of the Satellite Division of the Institute of Navigation, 12-15.
- [6] Zhou, Z., Hua, C., Fan, H. and Li, J., "Fault prediction of the nonlinear systems with uncertainty", Simulation Modelling Practice and Theory, 16 (6), 690-703. July.2008.
- [7] Cobano, J.A., Estremera, J. and Gonzalez de Santos, P., "Location of legged robots in outdoor environments", Robotics and Autonomous Systems, 56 (9), 751-761. Sep.2008.
- [8] Noureldin, A., El-Shafie, A. and Bayoumi, M., "GPS/INS integration utilizing dynamic neural networks for vehicular navigation", Information Fusion, 12 (1), 48-57. Jan.2011.
- [9] Linde, M., A Flocking and Obstacle Avoidance Algorithm for Mobile Robots, MS Thesis, Stockholm, Sweden, 2004.
- [10] Pears, N.E. and Bumby, J.R., "Guidance of an autonomous guided vehicle using low-level ultrasonic and odometry sensor systems", Transactions of the Institute of Measurement and Control, 11 (5), 231-248. Oct.1989.
- [11] Li, T-H S., Chang, C-C., Ye, Y-J., and Tasi, G-R., "Autonomous Parking Control Design for Car-Like Mobile Robot by Using Ultrasonic and Infrared Sensors", Lecture Notes in Computer Science, 4020, 472-479. 2000.
- [12] Soumare, S., Ohya, A. and Yuta, S., "Real-Time Obstacle Avoidance by an Autonomous Mobile Robot using an Active Vision Sensor and a Vertically Emitted Laser Slit", Intelligent Autonomous Systems, 7, 301-308. 2002.
- [13] Aitkenhead, M.J. and McDonald A.J.S., "A neural network based obstacle-navigation animat in a virtual environment", Engineering Applications of Artificial Intelligence, 15 (3-4), 229-239. Jun.2002.

- [14] Lee, K., Chung, W. and Yoo, K., "Kinematic parameter calibration of a car-like mobile robot to improve odometry accuracy", *Mechatronics*, 20 (5). 582-595. Aug/Sep.2010.
- [15] Chen, C. and Richardson, P., "Mobile robot obstacle avoidance using short memory: a dynamic recurrent neuro-fuzzy approach", *Transactions of the Institute of Measurement and Control*, 34 (2-3). 148-164. Apr/May2012.
- [16] Ibrahim, F. and Tascillo, A., "DGPS/INS Integration Using Neural Network Methodology", in *IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI 2000)*, 114-121.
- [17] Haykin, S., *Neural Networks, A Comprehensive Foundation*, Prentice Hall Inc., New Jersey, 1999.
- [18] Abulafya, N., *Neural Networks for System Identification and Control*, MSc Thesis, University of London, London, 1995.
- [19] Norgaard, M., Ravn, O., Poulsen, N. K., Hansen, L. K., *Neural Networks for Modeling and Control of Dynamic Systems, A Practitioner's Handbook*, Springer, 2000.
- [20] Norgaard, M., Ravn, O. and Poulsen, N. K. "NNSYSID- Toolbox for System Identification with Neural Networks", *Mathematical and Computer Modeling of Dynamical Systems*, 8 (1). 1-20. Aug.2002.
- [21] Tsakiri, M., "Integrated GPS and DR for Land Vehicle Navigation", in the *Proceeding of WALIS*, 22-24.
- [22] Chiang, K. W., Noureldin, A. and E-Sheimy, N., "Multisensor Integrating using Neuron Computing for Land-Vehicle Navigation", *GPS Solutions*, 6 (4). 209-218. March.2003.
- [23] Canan, S. and Akkaya, R., "An Artificial Neural Network Dead Reckoning/GPS Integrated Navigation System for Mobile Robot", in *International Symposium on Innovations in Intelligent Systems and Applications*, 330-333.
- [24] Mingyue C., Dihua S., Weining L., Min Z. and Xiaoyong L., "Adaptive Tracking and Obstacle Avoidance Control for Mobile Robots with Unknown Sliding", *International Journal of Advanced Robotic Systems*, 9. 171-184 .Nov.2012.