TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.7, July 2014, pp. 5324 ~ 5330
DOI: 10.11591/telkomnika.v12i7.5660

Sensor Fusion via Brain Emotional Learning for Ground Vehicle

Alvaro Vargas-Clara, Sangram Redkar*
Department of Engineering and Computing Systems, Arizona State University
7001 E. Williams Field Road, Mesa AZ 85212, United States
*Corresponding author, e-mail: sredkar@asu.edu

Abstract

In this work, the analysis of a filter consisting of the Brain Emotional Learning (BEL) algorithm is presented. The inner workings of the BEL filter are based on emotional learning model in mammalians brain. The BEL filter is implemented in simulation for the purpose of sensor fusion in a ground vehicle. In simulation, the signals from a Global Positioning System (GPS) and an Inertial Navigation System (INS) are integrated, in order to accurately track the trajectory of a ground vehicle around a track. The BEL filter is provided with some sensory signal and reward signal, subsequently the filter seeks to diminish noise from both sensing units, thus eliminating tracking error. A performance comparison between the BEL filter, and the more commonly utilized Kalman filter is presented. The BEL filter demonstrated robustness to uncertainties from the sensing units, it adapts quickly with dynamical change in the plant, and has small computational cost. The BEL filter demonstrated to be effective in sensor fusion.

Keywords: *Navigation; Emotional decision making; Sensor integration; Kalman Filter*

1. Introduction

The purpose of a navigation system in a vehicle is to determine its current location, velocity, and direction; in other words determine the state of the vehicle. This information is usually obtained from multiple sensors on the vehicle. The sensors commonly used are a GPS, and an INS.

A GPS is a sensor that provides positioning data relative to an earth-centered coordinate system. It uses at least 4 or more satellites with an unobstructed line of sight to calculate position, time, and velocity. GPS receivers can obtain signals from GPS satellites under any weather conditions, and anywhere on Earth. GPS are available for civilian and military applications. They are highly accurate in three-dimensional positioning. GPS position errors are bounded and are dependent on the availability of GPS satellites [1].

An INS sensor uses acceleration, and rotational sensors to continuously calculate position, orientation, and velocity. Although, its primary output is position relative to an earth-centered coordinate system. In contrast to a GPS sensor, the INS position errors are not bounded, and grow with time. In addition, the errors are dependent on the quality of its inertial sensors [1].

The integration of GPS and INS are in efforts to combat each of the sensing unit's weaknesses. For example, INS are initially given position and velocity information from another source, and subsequently it generates its own updated position and velocity by integrating information received from its inertial sensors. However, any small errors which arise in the measurement are integrated into gradually larger errors. By integrating the INS with a GPS, the GPS capability for online calibration and error estimation will help mitigate the INS integration drift. Conversely, in the event that there is an obstruction to the line of sight between vehicle and satellites, and the GPS is unable to perform. The INS can perform as the short-term backup when GPS signals are unavailable. Therefore, as GPS and INS have complementary characteristics, their implementation is considered in an integrated approach [2].

As a result, the navigation system utilizes the output signals from these sensors and integrates them to obtain more precise information about the vehicle's state. This process of integration is commonly referred to as sensor fusion. There are numerous methods to fuse INS and GPS, such as, loosely coupled or tightly coupled integration. In the majority of these

Received January 23, 2014; Revised March 18, 2014; Accepted April 4, 2014

designs GPS and INS integration filter is usually some form of a Kalman filter [1], [3-6]. In most cases, an extended Kalman filter is implemented with inertial errors as its state to obtain satisfactory performance. Kalman filter equations are optimal when sensor observations are unbiased with white noise. Also, there is a heavy computational cost in Kalman filter implementation, due to constant updating of Kalman gains.

In this paper, we present a BEL filter integration approach to achieve lower computational effort but with competitive performance measures compared to the more commonly used Kalman filter.

The paper is organized as follows. The sensor integration BEL filter is discussed in section 2. Implementation of the BEL filter and simulation setup is discussed in section 3. Simulations results are presented in section 3. Lastly, conclusions are made in the section 4.

2. Proposed Method

The proposed filter utilized for sensor fusion consists of the BEL model. BEL is a network model which simulates the brain emotional learning process of mammalian was developed by Balkenius & Moren [7, 8]. It is a computational model of the amygdala, Orbitofrontal Cortex (OFC), thalamus, and sensory input cortex, which are known to be responsible for emotional learning and processing.

Researchers in control have taken interest in utilizing this BEL model as a feedback controller. This is motivated by the fact that research in psychology, AI, and cognitive science identify the reciprocal influences of emotion and cognition [8]. This is motivated by the fact that research in psychology, AI, and cognitive science identify the reciprocal influences of emotion, cognition and decision making [8]. Therefore, Lucas et al. [9], first introduced the Brain Emotional Learning Based Intelligent Controller (BELBIC) which consisted of the BEL model but utilized as direct adaptive feedback control. BELBIC has been implemented in many engineering systems applications, such as, power system [10], aerospace launch vehicle [11], queue management [12], flight simulation servo system [13], and other uncertain nonlinear systems [14]. In all applications the BEL model demonstrated robustness to uncertainties, on-line adaptability, and small computational cost. However, there is not any research in implementing BEL as a filter for sensor fusion.

The inner working of BEL is an action generation system founded on sensory input and reward signal [14]. The emotional learning occurs primarily in the amygdala. The learning of the amygdala is given in the following equation:

$$\Delta G_a = \alpha S_i \max\left(0, \mathrm{Re}w - A\right) \tag{1}$$

Where G_a is the amygdala gain, α is the amygdala learning rate, S_i is the sensory input, *Rew* is the reward signal, and *A* is the amygdala output. The *max* term is for making the learning in the amygdala monotonic, implying that learning in amygdala should be permanent.

Similarly, the learning rule in OFC is shown in the following equation:

$$\Delta G_o = \beta S_i \left(MO - \mathrm{Re}w\right) \tag{2}$$

Where G_o is the OFC gain, β is the OFC learning rate, and *MO* is the model output, calculated as in Equation (3):

$$MO = A - O \tag{3}$$

In which, *O* is the output of the OFC. The model first receives the sensory input, S_i , then the model calculates the internal signals of the amygdala and OFC, these signals are calculated as in Equation (4) and (5):

$$A = G_a S_i \tag{4}$$

$$O = G_o S_i \tag{5}$$

The amygdala learns to predict and react to give an emotional signal. While the OFC system detects the difference between the expected system's prediction and the actual received emotional signal [15].

Controllers based on the BEL model demonstrated robustness to uncertainties, while being simple and having low computational cost. To utilize this version of the BEL model as a filter, it is important to understand that BEL model in essence converts two sets of inputs (S_i and *Rew*) into a decision signal as its output. Therefore, it is important to implement this BEL model in an appropriate manner so that input signals and output signals have the proper interpretations for the problem at hand.

3. Research Method

In this paper, a simulation of a ground vehicle around a track is utilized to draw performance comparison between Kalman Filter and the BEL filter. The performance of these two filter is based on their ability reduce noise from GPS as the vehicle trajectory is tracked. Two tracks are simulated, a circular and figure-8 track. The vehicle is modeled as traveling at a velocity of 5 m/s. The trajectory of the vehicle on the track is given by the following equation [1]:

$$\boldsymbol{\delta}_{pos} = \begin{bmatrix} Northing \\ Easting \\ -Down \end{bmatrix} = \begin{bmatrix} 3S\sin(\omega t+\phi) \\ 2S\sin(\omega t+\phi)\cos(\omega t+\phi) \\ -1/2h\cos(\omega t+\phi) \end{bmatrix} \tag{6}$$

Where S is the track scaling parameter, h is the crossover height, ω is mean angular speed, and ϕ is an arbitrary phase angle. This model is implemented in MATLAB, which also calculates vehicle velocity, acceleration attitude, and attitude rates. The trajectories simulated can be seen in Figure 1. Both simulated tracks have changes in elevation of 10 meters.

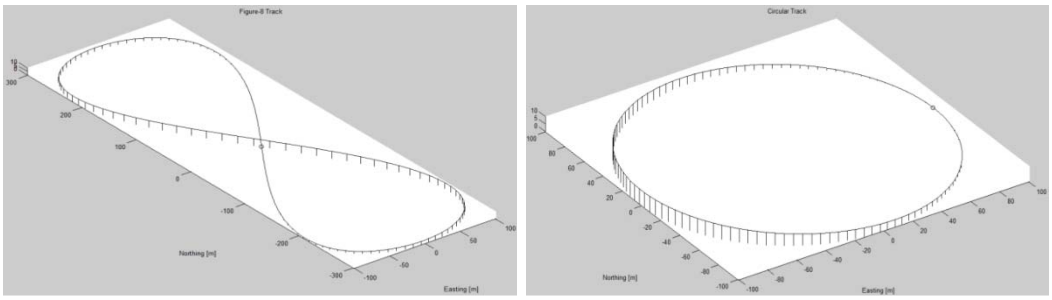

Figure 1. Figure-8 Track (left) and Circular Track (right)

The vehicle dynamic model consist of a Type2 Tracking Model. This tracking model can estimate position, velocity in three dimensions, given the appropriate measurements. The tracker utilizes a host vehicle dynamic model with zero-mean white noise acceleration, unbounded steady-state mean squared velocity and unbounded steady-state mean squared position variations. The full tracking model is implemented, which include three position components and three velocity components. The necessary Kalman filter components for a 3-dimension Type2 tracking filter are the following:

$$\mathbf{P_0} = \begin{bmatrix} \sigma_N^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_E^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_D^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v,N}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v,E}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v,D}^2 \end{bmatrix} \tag{7}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{acc}^2 \Delta t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{acc}^2 \Delta t^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{acc}^2 \Delta t^2 \end{bmatrix} \quad (9)$$

Where $\mathbf{P_0}$ is the estimation uncertainty covariance matrix, $\mathbf{\Phi}$ is the state-transition matrix, and $\mathbf{Q}$ is the covariance of dynamic disturbance noise.

The Kalman filter utilized for the performance comparison is of the following form:

$$\mathbf{K} = \mathbf{PH^T}\left[\mathbf{HPH^T} + \mathbf{R}\right] \quad (10)$$

Where $\mathbf{K}$ is the Kalman gain, $\mathbf{H}$ is measurement sensitivity matrix, and $\mathbf{R}$ is the sensor noise covariance matrix.

$$\mathbf{x}_1 = \mathbf{x}_1 + \mathbf{K}\left[\mathbf{z} - \mathbf{Hx}_1\right] \quad (11)$$

Where $\mathbf{z}$ is measurement vector, which is composed of the computed position, velocity and clock errors from the GPS.

$$\mathbf{P} = \mathbf{P} - \mathbf{KHP} \quad (12)$$

The implementation follows the above equations in chronological order. First, the Kalman gain is computed by Equation (10); Followed by the corrected state estimation in Equation (11); lastly, the corrected covariance matrix is computed by Equation (12). To finalize the Kalman filter implementation, the temporal updates are computed by the following equations:

$$\mathbf{x}_1 = \mathbf{\Phi x}_1 \quad (13)$$

$$\mathbf{P} = \mathbf{\Phi P \Phi^T} + \mathbf{Q} \quad (14)$$

The implementation of BEL model as a filter is chosen to be in similar manner as the Kalman filter implementation. This done in efforts to draw an accurate performance comparison between BEL filter and Kalman filter. However, slight differences arise due to the fact that BEL model is originally designed for descriptive purpose with no enigeering application in mind. Therefore, it is upto the designer to appropriately select the sensory input signal and reward signal in accordance to engineering application.

For the implementation of the BEL filter in this study, we selected the sensory input (S_i) to be of the form (15):

$$S_i = \mathbf{z} - \mathbf{x_1} \quad (15)$$

Where $\mathbf{x_1}$ is the vehicle states obtained from the vehicle trajectory model. *GPS* data can be obtain from a number of satellites, ranging from 4 to 29. In addition, GPS noise can be simulated to be of different noise distributions.

The reward function (*Rew*) is selected with objective of minimzing the difference between *GPS* and *Measured*. This function plays an important role in BEL filter. The filter attempts to increase the reward while minimizing the sensory input. The implemented reward function is given in Equation (16):

$$\mathrm{Re}w = -K_1 |S_i| + K_2 \tag{16}$$

Where K_1 and K_2 are gains. The reward function gains are positive real numbers. From Equation (16), it can be seen that BEL filter obtains maximum reward when the sensory input is zero. Closely noticing Equation (15), the sensory input is in essence, an error signal. The BEL filter tries to diminish the the error.

To carry out the simulation a number of parameters had to be selected. First, the learning rates for the amygdala and OFC were selected to be $\alpha = 1e-6$, and $\beta = 1e-4$, respectively. The OFC learning rate was chosen to be slightly larger to make the OFC learn the error in the amygdala quicker than the amygdala itself to eliminate the error. The other parameters were the gains in the Rew function, which were selected to be K_1 = 0.001 and K_2 = 1. These parameters and learning rates were selected through trial and error to improve BEL filter performance.

All simulations are carried out in MATLab. The number of satellites for GPS is varied. In addition, GPS noise distributions are varied. Performance measures for both Kalman and BEL filter are average RMS error for postions, velocity, and average Central Processing Unit (CPU) time.

4. Results and Analysis

The first scenario simulated is with a circular track. The simulation time is selected to be 0.2 hours. The first 100 seconds of the simulation data is not sampled to allow settling time. The simulation is executed 100 iterations. The number of satellites for this scenario is 29. Performance of Kalman and BEL filter are obtained, results are shown in Table 1.

Table 1. Performance Comparison for Circular Track Simulation

Kalman Filter							
GPS Noise Distribution	Avg. CPU Time [sec]	Avg. RMS Error					
		Position N [m]	Position E [m]	Position D [m]	Velocity N [m/s]	Velocity E [m/s]	Velocity D [m/s]
N(0,2)	1.145	10.316	7.779	5.373	3.206	3.333	0.101
U(-1,1)	1.268	8.131	6.045	4.772	3.031	3.228	0.092
Exp(2)	1.165	10.819	8.305	8.245	3.192	3.335	0.157
Tri(-1,0,1)	1.133	7.854	5.721	4.749	3.027	3.212	0.089
Wei(1,2)	1.174	8.258	5.951	5.750	3.026	3.219	0.105
BEL Filter							
GPS Noise Distribution	Avg. CPU Time [sec]	Avg. RMS Error					
		Position N [m]	Position E [m]	Position D [m]	Velocity N [m/s]	Velocity E [m/s]	Velocity D [m/s]
N(0,2)	0.586	4.915	5.017	4.934	4.908	4.998	4.919
U(-1,1)	0.575	2.276	2.861	2.714	2.768	2.868	2.710
Exp(2)	0.593	4.925	4.885	4.894	4.927	4.935	4.911
Tri(-1,0,1)	0.540	1.937	2.010	1.907	1.944	2.014	1.907
Wei(1,2)	0.598	4.264	4.018	4.169	4.353	4.250	4.235

The above table demonstrates that BEL filter was superior in diminishing positional errors. This trend was maintained through all GPS noise distributions. In some cases, it even performed better than Kalman filter in reducing velocity errors. A significant result obtained is that BEL performed better in reducing the computational cost across all noise distribution cases. In the worst case, BEL CPU time was half of the Kalman filter best CPU time.

The second scenario simulated is with a figure-8 track. This simulation was conducted in similar fashion as the first scenario. The figure-8 track simulated a more demanding tracking trajectory. Table 2 illustrates the results obtained from the second simulation scenario.

Table 2. Performance Comparison for Figure-8 Track Simulation

Kalman Filter							
GPS Noise Distribution	Avg. CPU Time [sec]	Avg. RMS Error					
		Position N [m]	Position E [m]	Position D [m]	Velocity N [m/s]	Velocity E [m/s]	Velocity D [m/s]
N(0,2)	1.195	11.259	8.241	5.481	1.421	1.233	0.100
U(-1,1)	1.190	9.113	6.413	4.932	1.032	0.879	0.090
Exp(2)	1.198	11.429	8.451	8.470	1.458	1.211	0.152
Tri(-1,0,1)	1.158	8.674	6.196	5.034	1.002	0.858	0.089
Wei(1,2)	1.202	8.829	6.201	5.837	0.998	0.864	0.109
BEL Filter							
GPS Noise Distribution	Avg. CPU Time [sec]	Avg. RMS Error					
		Position N [m]	Position E [m]	Position D [m]	Velocity N [m/s]	Velocity E [m/s]	Velocity D [m/s]
N(0,2)	0.588	4.936	5.035	4.929	4.915	5.006	4.932
U(-1,1)	0.581	2.823	2.822	2.796	2.830	2.837	2.799
Exp(2)	0.591	4.939	4.928	4.905	4.957	4.943	4.854
Tri(-1,0,1)	0.543	2.006	1.972	1.992	2.009	1.995	1.992
Wei(1,2)	0.594	4.264	4.092	4.180	4.375	4.158	4.267

Results obtained from the Figure-8 track simulation are similar to the ones obtained in the previous scenario, but with slightly higher CPU time and positional errors for both Kalman filter and BEL filter implementations. The figure-8 track appears to be no more rigorous than the circular track. For further performance comparison between the two filter implementations, a more interesting scenario is analyzed.

To conclude, the effects of the number of satellittes available is analyzed. As previously discussed, the number of satellites is a determental factor for GPS to accurately calculate position and velocity of a vehicle. Therefore, for this last scenario the number of satellites is varied from 4 to 29. Their effects on the Kalman and BEL filter performance are obtained, shown in Table 3.

Table 3. Effects of Number of Satellites on Kalman and BEL Filter Implementation

Kalman Filter: Normal Noise Distribution							
No. Sats	Avg. CPU Time [sec]	Avg. RMS Error					
		Position N [m]	Position E [m]	Position D [m]	Velocity N [m/s]	Velocity E [m/s]	Velocity D [m/s]
4	0.466	414.316	2694.149	258.880	2.096	6.439	0.075
9	0.620	29.285	30.007	29.869	1.831	1.695	0.129
14	0.767	14.058	8.515	6.582	1.712	1.258	0.106
19	0.920	12.259	8.546	6.077	1.527	1.258	0.105
24	1.011	12.051	8.294	5.774	1.519	1.242	0.104
29	1.195	11.259	8.241	5.481	1.421	1.233	0.100
BEL Filter: Normal Noise Distribution							
No. Sats	Avg. CPU Time [sec]	Avg. RMS Error					
		Position N [m]	Position E [m]	Position D [m]	Velocity N [m/s]	Velocity E [m/s]	Velocity D [m/s]
4	0.341	5.054	4.950	4.955	5.090	4.960	4.909
9	0.380	4.977	4.947	4.934	4.952	4.981	4.941
14	0.437	4.929	4.897	4.916	4.937	4.941	4.914
19	0.476	4.932	4.921	4.915	4.923	4.929	4.924
24	0.539	4.935	4.998	4.924	4.920	5.125	4.930
29	0.588	4.936	5.035	4.929	4.915	5.006	4.932

In the majority of the cases the RMS error for position and velocity increased as the number of satellites decreased for both filter implementations. However, the increments in the BEL implementation were small in comparison to the Kalman filter. In the Kalman filter implementation, the RMS error for position and velocity appear to grow exponentially when the satellites decreased from 14 to 4. The results demonstrate that the BEL filter is less sensitive to the effects of the number of satellites available. In addition, the CPU time increased as the number of satellites increased for both filter implementations. Although, this effect was more noticeable for the Kalman filter implementation. Lastly, a similar trend was obtained in that the BEL filter was superior at diminishing positional errors, while the Kalman filter was superior at reducing the velocity errors. An important note about this scenario, the effects on the number of satellites was carried out with a Gaussian GPS noise distribution.

The results from this study demonstrated the BEL qualities as a filter. It successfully filtered the noise from GPS and was able to accurately follow the trajectory of a vehicle around a track. It demonstrated robustness to a variety of noise distributions, and all this with significantly less computational cost.

5. Conclusion

For navigation, one of the most important task is to be able to accurately obtain the vehicle's state from an assortment of sensors. In this paper a new filter is developed, which is based on BEL model is investigated. In simulation, the BEL model is implemented as a filter in efforts to reduce GPS sensor noise and to accurately obtain vehicle's states as it is traveling around a track. The results from this study demonstrate the BEL qualities as a filter. It performed better at reducing positional RMS error while having significantly less computational cost than the traditional Kalman filter implementation. In addition, results show that BEL filter is less sensitive to the effects of the number of satellites available to accurately obtain GPS data. However, the BEL filter performance is greatly affected by the selection of the sensory input and reward signal. Further research in the characterization of the sensory input and reward signal can further enhance the BEL filter performance.

In conclusion, the BEL filter can be used in the real time application for filtering sensor noise on account of its robustness to noise uncertainty, and small computational cost.

References

[1] Grewal MS, Weill LR, Andrews AP. Global Positioning Systems, Inertial Navigation, and Integration. 2nd ed. New Jersey: John Wiley & Sons. 2007: 255-256.

[2] H Qi, JB Moore. Direct Kalman Filtering Approach for GPS/INS Integration. *IEEE Trans. Aerosp. Electron. Syst.* 2002; 38(2): 687–693.

[3] Wei M, Schwarz KP. A discussion of models for GPS/INS integration. In Y. Bock and N. Leppard (Eds.), Global Positioning Systems: An Overview. New York: Springer-Verlag. 1989: 316-327.

[4] Schwarz KP, Wei M, Gelderen MV. *Aided versus embedded: A comparison of two approaches to GPS/INS integration*. In Proceedings of IEEE PLANS. 1994; 314-321.

[5] B Guangrong, L Hongshen, H Ninghui. Multisensor Data Processing and Fusing Based on Kalman Filter. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(3): 1266-1277.

[6] H Guo. Neural Network Aided Kalman Filtering for Integrated GPS/INS Navigation System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(3): 1221-1226.

[7] J Moren, C Balkenius. *A Computational Model of Emotional Learning in the Amygdala*. in: From Animals to Animals 6, Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior. MIT Press. Cambridge. 2000.

[8] Balkenius C, Moren J. Emotional learning: A Computational Model of the Amygdala. *Cybernetics and Systems*. 2001; 32(6): 611–636.

[9] Lucas C, D Shahmirzadi, N Sheikholeslami. Introducing BELBIC: Brain Emotional Learning Based Intelligent Controller. *Int. J. Intell. Autom. Soft Comput*. 2004; 10(1): 11-22.

[10] Mohammdi MR, Lucas C, Najar AB. *A Novel Controller for a Power System based BELBIC*. In Proceedings of World Automation Congress. 2004; 409-420.

[11] Mehrabian AR, Lucas C, Roshanian J. Aerospace launch vehicle control: An intelligent adaptive approach. *Aerospace Science and Technology*. 2006; 149-155.

[12] Kharaajoo MJ. *Application of brain emotional learning based intelligent controller (BELBIC) to active queue* management. International Conference on Computational Science. 2004; 662-665.

[13] Ziyang Z, Daobo W, Zhisheng W. *Flight simulation servo system based on learning based intelligent controller*. Proceedings of the Fourth International Conference on Impulse and Hybrid Dynamic Systems. 2007; 1372-1375.

[14] Huang G, Zhen Z, Wang D. *Brain emotional learning based intelligent controller for nonlinear system*. Paper presented at Second international symposium on intelligent information technology application. 2008; 660-663.

[15] Mehrabian AR, Lucas C, Roshanian J. Design of an aerospace launch vehicle autopilot based on optimized emotional learning algorithm. *Cybernetics and Systems*. 2008; 39(3): 284-303.