Multi-Sensor Integration Based on a New Quantum Neural Network Model for Land-Vehicle Navigation

Debao Yuan*, Liangli Cai, Meng Li, Chen Liang, Xiaobo Hou

ABSTRACT

This paper aims to develop an efficient and accurate multi-sensor integration method for land vehicle navigation. For this purpose, a novel multi-sensor integration model was created based on quantum neural networks (QNNs) and back-propagation training. According to the information interaction mode of biological neurons and the theory on the QNNs, the author firstly put forward a QNN consisting of weighting, aggregation, activation and prompting, and then built a QNN model based on the proposed network. Then, the multi-layer feedforward QNN was combined with back-propagation learning to form a multi-sensor integration approach for land-vehicle navigation. Finally, the efficiency and accuracy of the proposed approach was verified through simulation and field test. This research sheds new light on the integration of data from multiple sensors and the improvement of land-vehicle navigation.

Key Words: Quantum Neural Networks (QNNs), Multi-data Integration, Quantum Neuron

DOI Number: 10.14704/nq.2018.16.6.1574 **NeuroQuantology 2018; 16(6):519-624**

Introduction

Quantum neural networks (QNNs) are neural network models based on the principles of quantum mechanics (Guptaa *et al.*, 2001; Purushothaman *et al.*, 1997; Behrman *et al.*, 2000; Toth *et al.*, 2000). The existing methods for QNN research fall into two categories: one improving the existing neural network through processing of quantum information, and the other exploring potential quantum effects in the brain (Satinover *et al.*, 2001; Jabbarpour *et al.*, 2002; Mould *et al.*, 2004). Focusing on QNN computing, researchers have combined artificial neural network (ANN) models with classic quantum theory into effective QNNs algorithms (Silva *et al.*, 2015; Panella *et al.*, 2011).

Most QNN learning algorithms follow the traditional model of an ANN to learn the input-output function of a given training set, and rely on traditional feedback mechanism to adjust parameters of the network until they converge to an optimal state (Rigatos *et al.*, 2006). Recent years saw the emergence of a new post-learning method that improves set of weights based on analogy with quantum effects occurring in nature. During the analogy, a biological neuron is simulated as a semiconductor heterostructure involving one energetic barrier sandwiched between two energetically lower areas (Kapanova *et al.*, 2017). Compared to the traditional model, this new algorithm can be achieved with the minimal additional computing costs.

The QNN research covers many computing perspectives rather than be limited to quantum effects in biological neural networks (Zhou *et al.*, 1999; Narayanan *et al.*, 2000; Gandhi *et al.*, 2014). For instance, much attention has been paid to multi-sensor integration, which aims

Corresponding author: Debao Yuan
Address: College of Geoscience and Surveying Engineering, Surveying Engineering, China University of Mining & Technology, Beijing 100083, China
e-mail ✉ yuandb@cumtb.edu.cn
Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
Received: 2 March 2018; **Accepted:** 21 April 2018

to collect data from different sensors for achieving the optimal navigation solution (Unsal *et al.*, 2002). Traditionally, the data from different sensors are combined by a filter through the process shown in Figure 1.

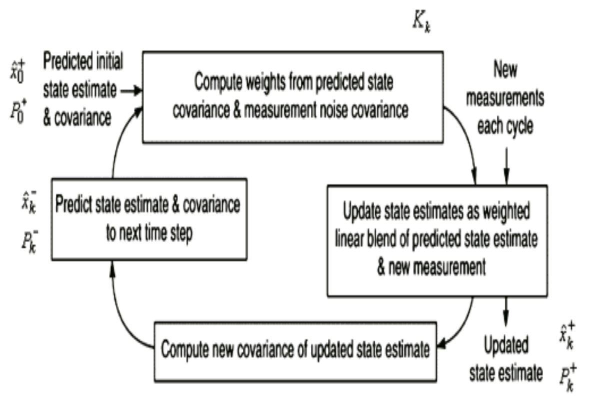

Figure 1. Traditional process of multi-sensor integration

The traditional process may contain errors from various sources. As shown in Figure 1, the initial error state $\hat{x}_0^+$ and the corresponding error covariance p_0^+ are assumed before the integration. Both are updated automatically when the filter enters the updating mode. In this way, the filter integrates all data together to give the optimal evaluation of the error state (Chen *et al.*, 1992). The above data integration process may apply to multi-sensor integration but has some drawbacks in land vehicle navigation. After all, it is immensely difficult to make optimal estimation of errors owing to the update of GPS position or velocity.

For efficient and accurate land vehicle navigation, this paper creates a multi-sensor integration model based on multi-layer feedforward QNN, which uses a back-propagation learning algorithm.

QNN model

QNN

ANNs are a family of computing models inspired by the biological neural networks in the human brain. These networks are aggregated by identical units called neurons (Schuld *et al.*, 2015). Figure 2 illustrates the basic model of the neurons.

Each neuron consists of three parts: the weight link w_{kj}, the activation function $\varphi(\cdot)$ for neuron output, and the final output y_k. Among them, the weight link adds up all input values weighted through neuron (v_k) and external bias (b_k).

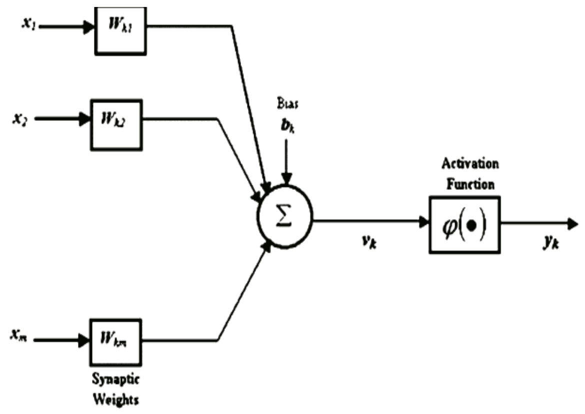

Figure 2. Basic model of neurons

The proposed QNN model (Figure 3) covers weighting, aggregation, activation and prompting. Specifically, weighting simulates the bond strength between synapses in the neuron; aggregation simulates the time and space integration of stimuli received by multiple synapses; activation simulates the interaction effect of the membrane potential changes and the active value of the neurons; prompting simulates the nonlinear features of neurons like excitation, inhibition, fatigue, refractory period and threshold.

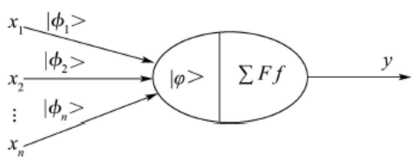

Figure 3. The proposed QNN

Note: $|\phi_i>$ is the weight of x_i, $|\varphi>$ is the active value, Σ is an aggregation operator, F is the activation function, and f is the inspirit function.

In the proposed model, $X = (x_1, x_2, \dots x_n)^T$ is the input vector; y is the real-valued output; $|\phi>= (|\phi_1>, |\phi_2>, \dots |\phi_n>)^T$ is the weight, with $|\phi_i>= (cos\theta_i, sin\theta_i)^T, i = 1,2, \dots n$; $|\varphi>= (cos\xi, sin\xi)^T$ is the active value; τ is the threshold.

The input-output relationship of the QNN model can be expressed as:

$$y = f(F(X^T\,|\phi>, |\varphi>) - \tau)$$
$$= f\left(\sum_{i=1}^{n} x_i < \phi_i|\varphi> - \tau\right)$$
$$= f(\sum_{i=1}^{n} x_i \cos(\theta_i - \xi) - \tau)$$

where F is the inner product operator; f is the sigmoid function.

QNN model

According to the proposed QNN model, a three-layer feedforward QNN model was established with a certain number of neurons (Figure 4). The three layers are the input layer, the middle layer and the output layer.

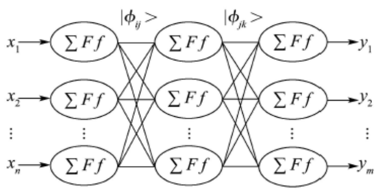

Figure 4. The proposed QNN model

Note: n, p and m are respectively the number of quantum neurons in the input layer, the middle layer and the output layer; x_i is the network input; h_j is the hidden layer output; y_k is the network output; $|\phi_{ij}>$ is the hidden layer weight; $|\varphi_j>$ is the active value of the middle layer; τ_j is the hidden layer threshold; $|\phi_{jk}>$ is the output layer weight; $|\varphi_k>$ is the active value of the output layer; τ_k is the output layer threshold.

The input-output relationship of the proposed QNN model can be expressed as:

$$y_k = f\left(\sum_{j=1}^{p} h_j < \phi_{jk}|\varphi_k > -\tau_k\right)$$

$$= f\left(\sum_{j=1}^{p} f\left(\sum_{i=1}^{n} x_i < \phi_{ij}|\varphi_j > -\tau_j\right) < \phi_{jk}|\varphi_k > -\tau_k\right)$$

$$= f\left(\sum_{j=1}^{p} f\left(\sum_{i=1}^{n} x_i \cos(\theta_{ij} - \xi_i) - \tau_j\right) \cos(\theta_{jk} - \xi_k) - \tau_k\right)$$

where $i = 1, \ldots, n, j = 1, \ldots, p, k = 1, \ldots, m$.

Multi-Sensor Integration Based on QNN Model

Three-layer QNN model with back-propagation learning

As mentioned above, the proposed feedforward QNN model consists of the input layer, the hidden layer, and the output layer. The numerous input layer neurons are responsible for receiving the input; the output layer neurons are responsible for delivering the output to the user; the middle layer, through its nonlinear activation function, enables the QNN to solve nonlinear problems.

The proposed QNN model can integrate the data from multiple sensors. On this basis, the model was trained with a back-propagation mechanism for application in land-vehicle application. The trained model is displayed in Figure 5. The output of the model is the disparity between the current and previous positions of the inertial navigation system.

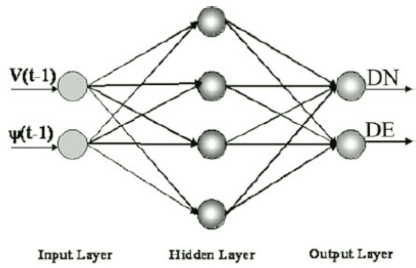

Figure 5. QNN trained for land-vehicle navigation

Note: V(t–1) is the velocity of the inertial navigation system; $\psi(t-1)$ is the direction of the inertial navigation system; DN(t) and DE(t) are the current and previous positions of the system, respectively.

Design and training of the QNN for land-vehicle navigation

The QNN can be trained through learning the input signal and expected output of training samples, without needing any prior knowledge. The signal propagation in the training of the QNN is presented in Figure 6.

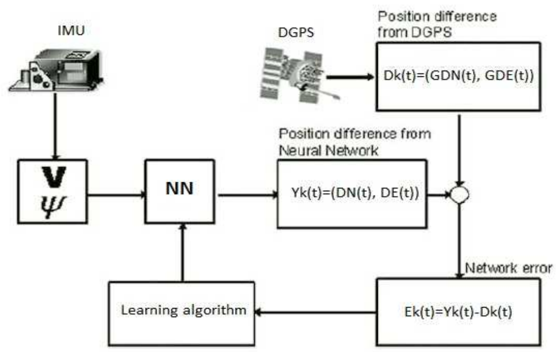

Figure 6. Signal propagation in the QNN

The network error $E_k(t)$ can be obtained as:

$$E_k(t) = D_k(t) - Y_k(t)$$

where $Y_k(t)[DN(t), EN(t)]$ is the final output of the QNN; $D_k(t) = [GDN(t),\ EDN(t)]$ is desired output. The network error lays the basis for adjusting the weights of neurons.

In the training process, the weight and active value of quantum correction can be obtained by the single bit quantum logic gates. The correction formula can be expressed as:

$$|\phi(t+1)> = \begin{bmatrix} \cos\Delta\theta_t & -\sin\Delta\theta_t \\ \sin\Delta\theta_t & \cos\Delta\theta_t \end{bmatrix} |\phi(t)>$$

$$|\varphi(t+1)> = \begin{bmatrix} \cos\Delta\xi_t & -\sin\Delta\xi_t \\ \sin\Delta\xi_t & \cos\Delta\xi_t \end{bmatrix} |\varphi(t)>$$

The core idea of the algorithm is to set the rotation angle of the quantum logic gates $\Delta\theta(t)$ and $\Delta\xi(t)$ and modified amount of threshold $\Delta\tau(t)$, such that all three parameters can converge rapidly through iteration:

$$\theta(t+1) = \theta(t) + \Delta\theta(t)$$
$$\xi(t+1) = \xi(t) + \Delta\xi(t)$$
$$\tau(t+1) = \tau(t) + \Delta\tau(t)$$

The error signal $E_k(t)$ can be acquired by forward computation, while the backward computation of gradient δ can be expressed as:

$$\delta_j^{(\zeta)} = \begin{cases} E_j(t)\dfrac{d\varphi(v_j^L)}{dt} & \textbf{\textit{for neurons j in output layer L}} \\ \dfrac{d\varphi(v_j^L(t))}{dt}\sum_k \delta_k^{(\zeta-1)}(t) w_{kj}^{(\zeta-1)}(t) & \textbf{\textit{for neruons j in hidden layer}}\ \zeta \end{cases}$$

The weights can be calculated by the following rules:

$$\Delta w_{ji}(n) - \eta\delta_j^{(\zeta)}(n) y_i^{(\zeta-1)}(n) + \alpha\Delta w_{ji}^{\zeta}(n-1)$$
$$w_{ji}^{(\zeta)}(n+1) - w_{ji}^{(\zeta)}(n) + \Delta w_{ji}^{(\zeta)}(n)$$

Methods

Two experiments were carried out, including a simulation and a field test.

Simulation

To validate the proposed QNN model, two road conditions were simulated, including a smooth road conditions (Figure 7) and a harsh road condition (Figure 8).

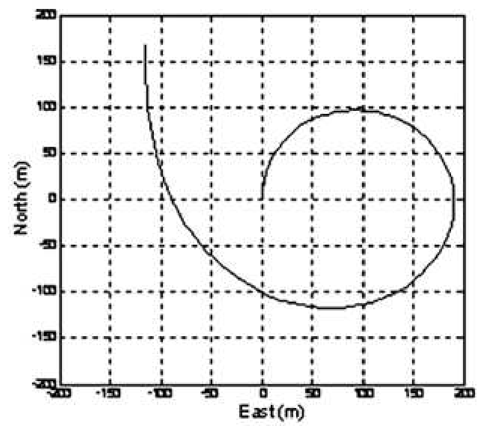

Figure 7. The smooth motion trajectory

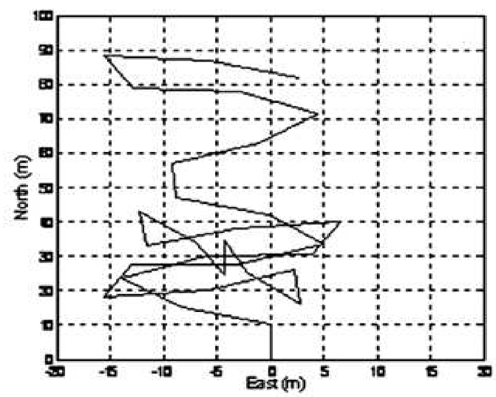

Figure 8. The harsh motion trajectory

Then, a 512-neuron QNN was adopted to simulate the smooth trajectory and the harsh trajectory. To verify the accuracy, the number of neurons in the middle layer was increased from 16 to 1,024. The prediction accuracy of the output should increase with the number of neurons. The predicted outputs for the smooth and harsh motion trajectories are shown in Figure 9 and Figure 10, respectively.

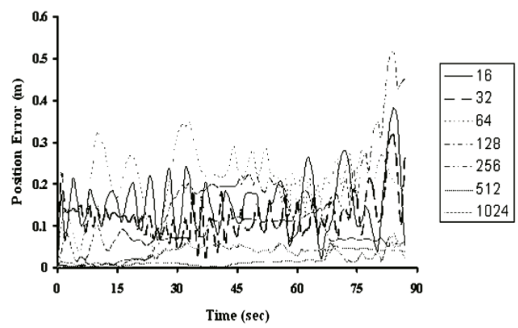

Figure 9. Predicted output for smooth motion trajectory

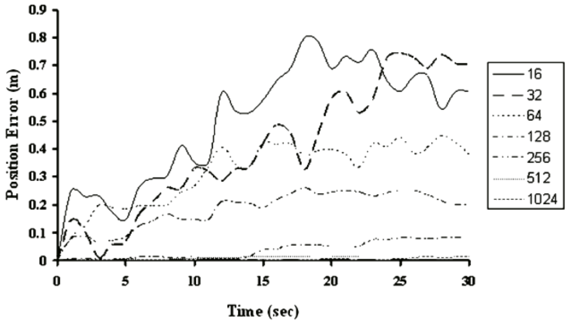

Figure 10. Predicted output for harsh motion trajectory

As shown in the two figures above, the number of position errors was reduced with the addition of neurons. However, the overflow of neurons may cause the over-constrained problem and thus lower the reliability of the output. Overall, the 512 middle layer neurons of the proposed model provided acceptable accuracy in the training.

Next, the standard deviation was simulated to identify the effect of random noises. The parameters of the simulation cases are listed in Table 1. The effects of velocity and direction errors on position errors are respectively displayed in Figure 11 and Figure 12.

Table 1. Parameters of simulation cases

Cases	Velocity errors(**m/s**)				Heading errors		
	A	B	C	D	a	b	c
Standard deviation	1	10	50	100	1	3	5

From Figures 11 and 12, it can be seen that the proposed QNN network was less sensitive to velocity errors than direction errors. Hence, the effect of direction errors are more significant than that of velocity errors.

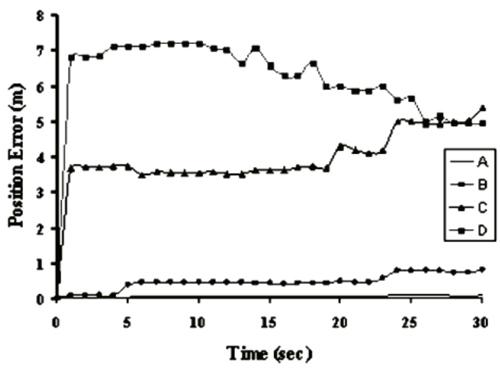

Figure 11. Influence of velocity errors on position errors

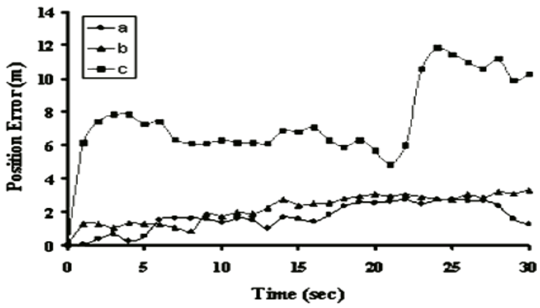

Figure 12. Influence of direction errors on position errors

Field test

This subsection is about the field test performed to verify the performance of the proposed QNN model. The 2,600s-long field test contains two phases. In the first 2,200s, both the inertial navigation system and the global positioning system measurements were available and the QNN model was in the training mode. In the last 400s, the QNN model ran in the prediction mode.

The real trajectory and the trajectory generated by the QNN model are shown in Figure 13 and Figure 14, respectively. Comparing the two trajectories, it is clear that our QNN model has achieved a good efficiency and accuracy.

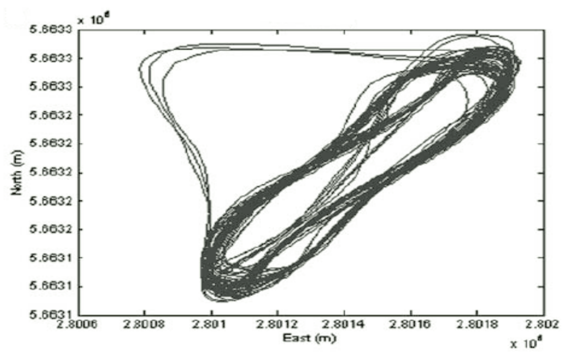

Figure 13. The real trajectory

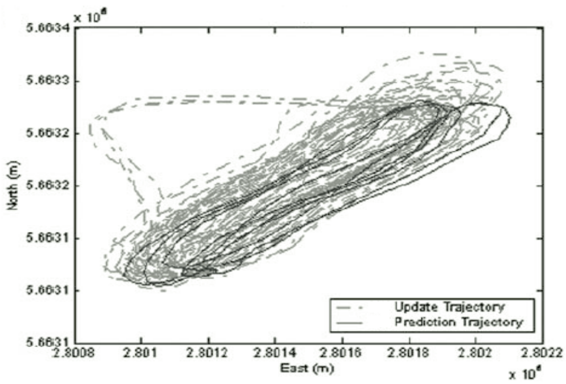

Figure 14. The trajectory generated by the QNN model

Conclusions

This paper proposes a multi-layer feedforward QNN with back-propagation learning, and applies it to integrate multi-sensor data for land-vehicle navigation. Then, the proposed QNN model was tested through simulation and field test. The experimental results show that our model can integrate multi-sensor data for automatic navigation and achieve a high efficiency and accuracy through training.

References

Behera L, Kar I, Elitzur AC. A Recurrent quantum neural network model to describe eye tracking of moving targets. Foundations of Physics Letters 2005; 18 (4): 357-70.

Behrman EC, Nash LR, Steck JE. Simulations of quantum neural networks. Information Sciences 2000; 128(3): 257-69.

Chen G. Introduction to random signals and applied Kalman filtering. International Journal of Adaptive Control & Signal Processing 1992; 6 (5): 516-18.

Gandhi V, Prasad G, Coyle D. Quantum neural network-based EEG filtering for a brain–computer interface. IEEE transactions on neural networks and learning systems 2014; 25 (2): 278-88.

Guptaa S, Ziab RKP. Quantum neural networks. Journal of Computer & System Sciences 2001; 63 (3): 355-83.

Jabbarpour YM. The Quantum brain: The search for freedom and the next generation of man. Psychiatric Services 2002; 53 (9): 1185-86.

Kapanova KG, Dimov I, Sellier JM. On randomization of neural networks as a form of post-learning strategy. Soft Computing 2017; 21 (9): 2385-93.

Mould RA. Quantum brain rules. Christianity Today 2004;33 (4): 591-612.

Narayanan A, Menneer T. Quantum artificial neural network architectures and components. Information Sciences 2000; 128 (3-4): 231-55.

Panella M, Martinelli G. Neural networks with quantum architecture and quantum learning. International Journal of Circuit Theory & Applications 2011; 39 (1): 61-77.

Purushothaman G, Karayiannis N B. Quantum neural networks (QNNs): inherently fuzzy feed forward neural networks. IEEE Transactions on Neural Networks 1997; 8 (3): 679-93.

Rigatos GG, Tzafestas SG. Quantum learning for neural associative memories. Fuzzy Sets & Systems 2006; 157 (13): 1797-1813.

Satinover B, Jeffrey. The quantum brain. American Spectator 2001; 53 (9): 1185-86.

Schuld M, Sinayskiy I, Petruccione F. Simulating a perceptron on a quantum computer. Physics Letters A 2015; 379(7): 660-63.

Silva AJD, Oliveira WRD, Ludermir TB. Comments on “quantum M-P neural network”. International Journal of Theoretical Physics 2015; 54 (6): 1878-81.

Sun W, He YJ, Meng M. A Novel quantum neural network model with variable selection for short term load forecasting. Applied Mechanics & Materials 2010; 23: 612-17.

Toth G, Lent CS, Tougaw PD. Quantum cellular neural networks. Superlattices & Microstructures 2000; 20 (4): 473-78.

Unsal C, Kachroo P, Bay JS. Multiple stochastic learning automata for vehicle path control in an automated highway system. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 2002; 29 (1): 120-28.

Zhou J, Gan Q, Krzyżak A. Recognition of handwritten numerals by quantum neural network with fuzzy features. International journal on document analysis and recognition 1999; 2(1): 30-36.