https://ntrs.nasa.gov/search.jsp?R=20170009832 2020-05-09T22:38:04+00:00Z

UAV Trajectory Modeling Using Neural Networks

Min Xue*

NASA Ames Research Center, Moffett Field, CA 94035

Massive small unmanned aerial vehicles are envisioned to operate in the near future. While there are lots of research problems need to be addressed before dense operations can happen, trajectory modeling remains as one of the keys to understand and develop policies, regulations, and requirements for safe and efficient unmanned aerial vehicle operations. The fidelity requirement of a small unmanned vehicle trajectory model is high because these vehicles are sensitive to winds due to their small size and low operational altitude. Both vehicle control systems and dynamic models are needed for trajectory modeling, which makes the modeling a great challenge, especially considering the fact that manufactures are not willing to share their control systems. This work proposed to use a neural network approach for modelling small unmanned vehicle's trajectory without knowing its control system and bypassing exhaustive efforts for aerodynamic parameter identification. As a proof of concept, instead of collecting data from flight tests, this work used the trajectory data generated by a mathematical vehicle model for training and testing the neural network. The results showed great promise because the trained neural network can predict 4D trajectories accurately, and prediction errors were less than 2.0 meters in both temporal and spatial dimensions.

I. Introduction

A large amount of small Unmanned Aerial Vehicles (sUAVs) are expected to operate in the uncontrolled Class G airspace, which is at or below 500 feet Above Ground Level (AGL), where many static and dynamic constraints exist, such as ground properties and terrain, restricted areas, various wind conditions, manned aircraft operations, and collision avoidance among sUAVs. How to enable safe, efficient, and massive sUAV operations at the low altitude airspace remains a great challenge. NASA's Unmanned aircraft system Traffic Management (UTM) research initiative[1,2] works on establishing infrastructure and developing policies, requirement, and rules to enable safe and efficient sUAVs' operations. While there are lots of research problems need to be addressed, trajectory modeling should serve as the foundation for understanding sUAV operations and developing requirements and rules for UTM system.

For manned and unmanned large fixed-wing aircraft, where precision requirements are typically at a nautical-mile level, trajectory errors are dominated by navigation and sensor errors. Despite different controller designs, aircraft control systems are capable of maintaining trajectory deviation at a level that is much lower than a nautical-mile. Especially for en-route traffic, cross-track trajectory errors caused by winds are usually negligible and modeling aircraft controllers becomes unnecessary. This approach greatly simplifies trajectory models for large size fixed-wing aircraft,[3,4] especially when calculating horizontal trajectories. Horizontal trajectories of large size fixed-wing aircraft are usually constructed from straight lines and turn arcs, where turn radii are decided based on bank angles and ground speeds.[5–7] The simplified horizontal trajectory calculation has been widely used in aviation research.[8–11]

While, for sUAV, navigation errors are mitigated through technologies like Differential Global Positioning System (DGPS),[12,13] trajectory errors caused by wind rise because sUAVs are sensitive to wind due to their small size and low operational altitude. Different sUAV controllers usually produce different trajectory responses under the same wind condition and errors and differences are not negligible at the meter-level precision. Therefore, accurate models of sUAV control systems become as critical as dynamic systems when assessing and managing safe and efficient UTM operations. In the past, plenty of research[14–17] has been done from a vehicle designer's perspective - modeling sUAV's plant/dynamics through system identification and

*Aerospace Research Engineer, Aviation Systems Division. Mail Stop 210-8. AIAA senior member.

designing control system based on that. However, because manufacturers usually are not willing to share their control designs as a part of intellectual properties, how to model sUAV trajectory with unknown control systems becomes a challenging research question. On the other hand, considering the rapidly growing small UAV market, the low manufacturing cost, and the short length of manufacturing cycle for small UAVs, it might not be feasible to model a sUAV trajectory following the traditional way in manned aviation, where manufacturers provide aerodynamics based on their numerous wind tunnel and flight tests in design phases and trajectory models are usually accurate enough without knowing aircraft's control systems[a]. Because of the associated costs, efforts, and time, sUAV manufacturers may not conduct enough wind tunnel and flight tests as their conventional aircraft counterparts. This work proposed to use a neural network approach for modelling small unmanned vehicle's trajectory without knowing its control system and bypassing exhaustive efforts for aerodynamic parameter identification. A neural network based approach is proposed in this work to model sUAV's trajectory, where the entire vehicle model including dynamics and control systems is treated as a black box. As a proof of concept, instead of collecting data from flight tests, this work used the trajectory data generated by a mathematical vehicle model for training and testing. The neural network is first trained to learn the vehicle's responses at multiple conditions with identified features. Once being fully trained, given current vehicle states, winds, and desired future trajectory, the neural network will be used to predict the vehicle's future states at next time step. By repeating this step, a complete 4-D trajectory is then generated as time progresses.

In this work, Section II describes the method proposed for modeling vehicle dynamics and controls. Section III presents the experiments conducted for initial testing of the proposed neural network method. Section IV draws the conclusions.

II. Method

This section presents a generalized small UAV vehicle model and proposes a neural network based approach to model sUAV's trajectory.

A. Small UAV vehicle model

A nonlinear dynamic model for sUAVs, both multi-copters and fixed-wing aircraft, can be derived using Newton-Euler equations if the vehicle is assumed to be rigid and symmetric.[17,18] In the vehicle body axis, the model can be written as force, kinematic, and moment equations shown in Eqn. 1, 2, and 3, respectively. (u, v, w) are body axis velocities; (p, q, r) are body axis angular velocity rates; And (ϕ, θ, ψ) are roll, pitch, yaw angles in body axis, respectively. (F_x, F_y, F_z) are body axis aerodynamic forces and $(M_\phi, M_\theta, M_\psi)$ are body axis moments. J_x, J_y, and J_z are the moments of inertia about the principle axes in the body frame. Then the vehicle location in the Earth frame can be expressed as Eqn. 4, where p_n, p_e, and h are the north position, east position, and altitude, respectively, and s and c denote sin and cos for simplicity. Although the overall model can be generalized, models of aerodynamic forces (F_x, F_y, and F_z) and moments (M_ϕ, M_θ, and M_ψ) can be quite different because of differences in vehicle types.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -g\sin\theta \\ g\cos\theta\sin\phi \\ g\cos\theta\cos\phi \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} M_\phi \\ \frac{1}{J_y} M_\theta \\ \frac{1}{J_z} M_\psi \end{bmatrix} \tag{3}$$

[a]Difficulty arises when modeling departure and arrival trajectories because of unknown FMS settings.

$$\begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4}$$

B. Neural network approach

Multi-layered neural networks are universal approximators and they are capable of approximating any continuous function over a specified compact set, according to past literatures in neural network studies.[19–21] Various neural networks had been applied to nonlinear system identification and parameter estimation and showed a lot of promise.[22–24] Considering the challenge associated with collecting vehicle states, this work proposed to apply neural networks to model an entire vehicle including the plant and control system (shown in Fig. 1) to reduce the number of vehicle states required to be collected. In this approach, at each time step, the sUAV's current states($\vec{x_0}$, $\dot{\vec{x_0}}$), next flight states($\vec{x}$, $\dot{\vec{x}}$), and wind conditions ($\vec{w_x}$, $\vec{w_y}$) are used as training input and output pairs for the neural network. For a given vehicle, states are collected at various flight conditions. Once the training is finished, this neural network will be applied to predict future flight states at next time step based on current flight states. Therefore, one neural network should be capable of representing one vehicle model for predicting trajectories.

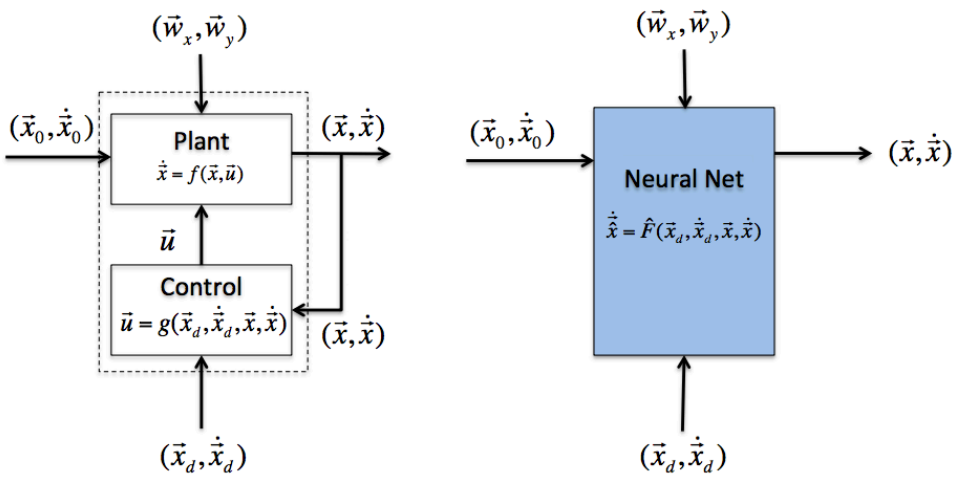


Figure 1. Trajectory modeling using Neural Networks

Because the model is used for trajectory prediction, inputs and outputs are limited to positions and speeds. Nine inputs are selected for trajectory calculation: position deviations, speed deviations, and speeds. Six defined outputs are position change rates and speed change rates. This setting leads to a multi-layered neural network composed of nine input neurons, 20 hidden neurons, and six output neurons as shown in Fig. 2.

The neural network is trained using the back-propagation approach with learning rate adaptation. The learning rate update rule followed Jacobs' hybrid algorithm,[25] in which a momentum[26] was added into the Delta-Bar-Delta rule. The weight w_{ij} between the ith and jth neurons is updated by Eqn. 5, where $E(k)$ is the mean square sum of the differences between the outputs and the desired target values and $\alpha = 0.75$.

$$\Delta w_{ij}(k+1) = \alpha \Delta w_{ij}(k) - \eta_{k+1} \frac{\partial E(k)}{\partial w_{ij}(k)} \tag{5}$$

And the learning rate η_{k+1} is updated by:

$$\eta_{k+1} = \eta_k + \Delta\eta_k \tag{6}$$

Where the $\Delta\eta_k$ is given by:

$$\Delta\eta_k = \begin{cases} \kappa, & \text{if } \bar{\delta}_{k-1} \cdot \delta_k > 0 \\ -\phi \cdot \eta_k & \text{if } \bar{\delta}_{k-1} \cdot \delta_k < 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

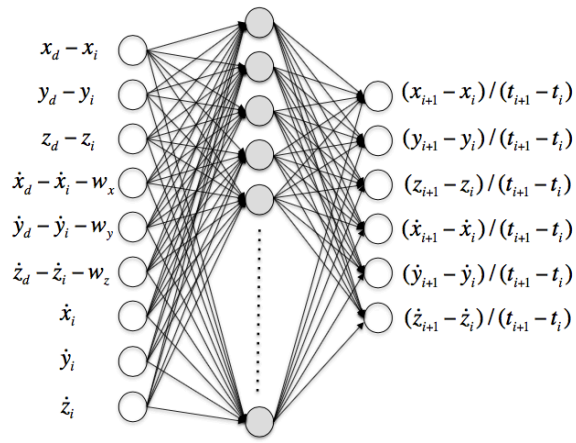


Figure 2. Neural network structure: (9 input units, 20 hidden units, and 6 output units)

The δ and $\bar{\delta}$ are govened by:

$$\delta_k = \frac{\partial E(k)}{\partial w_{ij}(k)} \tag{8}$$

and

$$\bar{\delta}_k = (1-\theta)\delta_k + \theta\bar{\delta}_{k-1} \tag{9}$$

where $\kappa = 0.01$, $\eta_0 = 0.75$, $\phi = 0.2$, and $\theta = 0.7$.

III. Experiments

Because the scope of this paper is limited to concept proof, trajectory data are generated by a mathematical vehicle model instead of being collected from actual flight tests. The generated trajectories are used as both training and testing data for the neural network. Experiments are set up to examine if a neural network can be trained to fit the training trajectory data and to check if the trained neural network can calculated testing trajectories in a sequential manner without diverging away from the truth.

A. Trajectory data generation

In order to generate trajectory data for training and testing, a small UAV dynamic model and a controller model are needed. Without loss of generality, a quadrotor model and a PD controller are selected for generating trajectory data sets. By neglecting Coriolis terms and small angle approximations, a quadrotor dynamics[18,27,28] can be expressed as Eqn. 10, where w_n and w_e are north and east components of the wind vector and vetical wind component is negelected here. The v_n and v_e are vehicle velocities in the Earth frame. The force and moments generated by motors can be simplified to Eqn. 11, where k_f and k_m are the aerodynamic force and moment coefficients for motors. L is the arm length and Ω_i is angular velocity of rotor i.

$$\begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{v_n} \\ \dot{v_e} \\ \ddot{h} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v_n + w_n \\ v_e + w_e \\ -(c\phi s\theta c\psi + s\phi s\psi)\cdot F_z/m \\ (-c\phi s\theta s\psi + s\phi c\psi)\cdot F_z/m \\ g - c\phi c\theta \cdot F_z/m \\ \frac{1}{J_x}M_\phi \\ \frac{1}{J_y}M_\theta \\ \frac{1}{J_z}M_\psi \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} F_z \\ M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} k_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ (-k_f\Omega_2^2 + k_f\Omega_4^2)\cdot L \\ (k_f\Omega_1^2 - k_f\Omega_3^2))\cdot L \\ (k_m\Omega_1^2 - k_m\Omega_2^2 + k_m\Omega_3^2 - k_m\Omega_4^2))\cdot L \end{bmatrix} \tag{11}$$

The PD position controller is shown in Eqn. 12 and Eqn. 13, where (x_d, y_d) and ($\dot{x_d}$, $\dot{y_d}$) are desired positions and velocities and k_p and k_d are associated gains. Eqn. 12 computes desired accelarations, and the desired roll and pitch angles (ϕ_d and θ_d) are then calculated using Eqn. 13. Eqn. 14 shows the control law for both roll and pitch angles, where $k_{p,\phi}$, $k_{d,\phi}$, $k_{p,\theta}$, and $k_{d,\theta}$ are gains.

$$\begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \end{bmatrix} = \begin{bmatrix} k_p(x_d - x) + k_d(\dot{x}_d - \dot{x}) \\ k_p(y_d - y) + k_d(\dot{y}_d - \dot{y}) \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \frac{m}{F_z} \begin{bmatrix} -\sin\psi & -\cos\psi \\ \cos\psi & -\sin\psi \end{bmatrix}^{-1} \begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} M_\phi \\ M_\theta \end{bmatrix} = \begin{bmatrix} k_{p,\phi}(\phi_d - \phi) + k_{d,\phi}(\dot{\phi}_d - \dot{\phi}) \\ k_{p,\theta}(\theta_d - \theta) + k_{d,\theta}(\dot{\theta}_d - \dot{\theta}) \end{bmatrix} L \tag{14}$$

This mathematical model was applied to generate trajectory data for training. As an initial effort, the vehicle's horizontal trajectory responses to side wind gust were generated: The vehicle is required to follow a straight line at a constant altitude with various ground speeds and it will enter a constant north wind field, where the wind direction is perpendicular to the flying direction. A total of 35 trajectories were generated by changing vehicle's ground speed and wind speed as shown in Table 1 and flight states were recorded every 0.01s. The current flight states and next flight states at each time step are treated as an input and output pair, respectively. There are a total of 62,850 pairs of training data from these trajectories.

Table 1. Parameters for trajectory generation

Parameter	Values	Units
Vehicle ground speed	2, 5, 8, 11, 12, 13, 14, 15	m/s
Wind speed	selected values in [1.0 9.5]	m/s

B. Neural network training

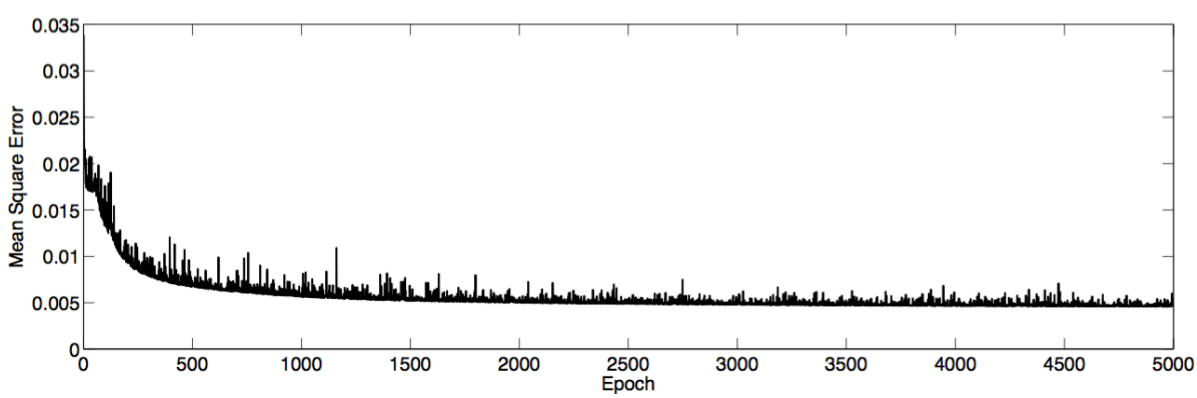


Figure 3. Mean square error in the training phase

In the training process, the training and testing data were applied without specfic ordering and the objective was to reduce overall MSE even though errors at certain data points may not be small. The neural

network was trained in 20,000 epochs and Fig. 3 shows the learning evolution for the first 5,000 epochs, where the MSE of the normalized outputs was reduced below 0.005. The MSE was ended at 0.0032 after 20,000 epochs. Fig. 4 presents the histogram of x position training errors. These errors are between unnormalized target outputs and the values generated by the trained neural network. The figure shows that most of the x position training errors are within ±0.2 meters. Fig. 5 presents error histograms of four outputs: x position, y position, forward speed (at x axis), and lateral speed (at y axis). It is noted that training errors of velocities are much higher compared with position errors; sometime they even exceed the ±2 meter per second range.

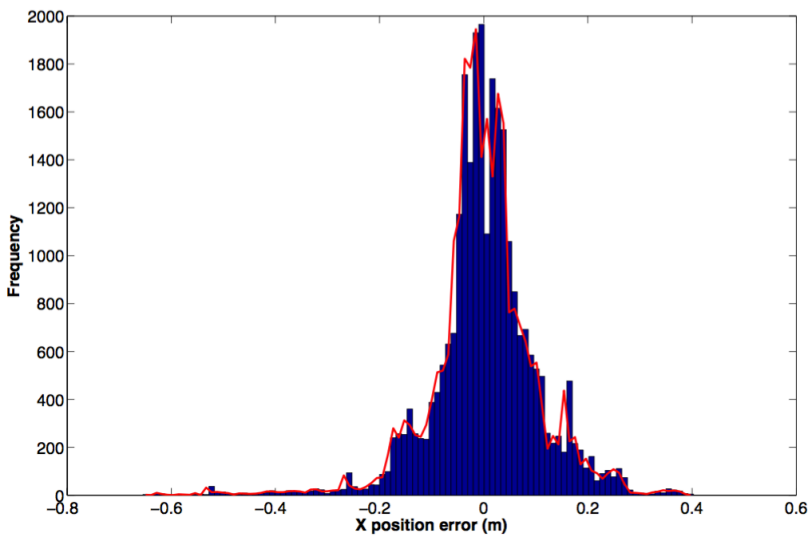


Figure 4. Histogram of errors in x position

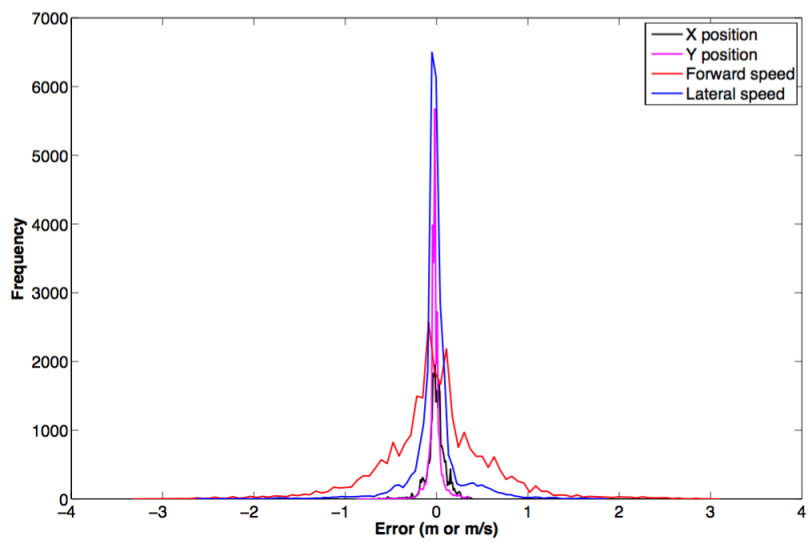
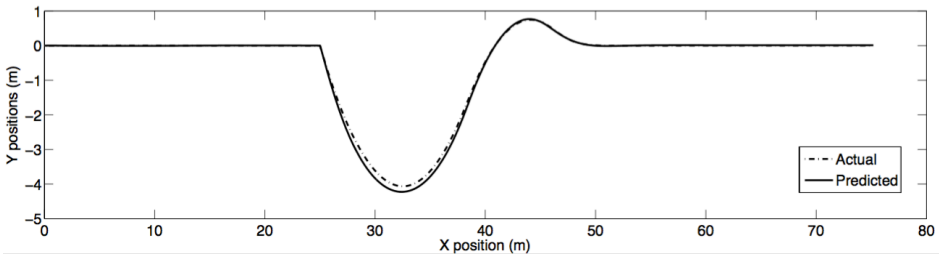


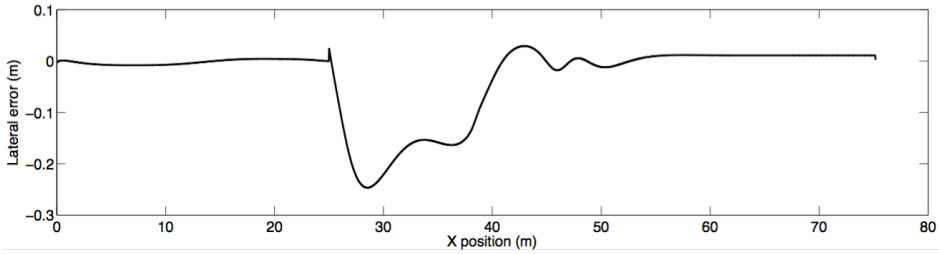
Figure 5. Histogram of errors in x position, y position, forward speed, and lateral speed

C. Trajectory prediction

In the process of predicting trajectories, the calculations are performed sequentially starting from the initial flight state. In the prediction experiment presented in this section, the vehicle's ground speed was set to 5m/s and the wind field was assumed to affect the area where $x >= 25m$. The wind came from the North and the vehicle flied from the West to the East. The wind speed was set to 5m/s. The experiment set-up is similar to the training cases although the values are not the same.

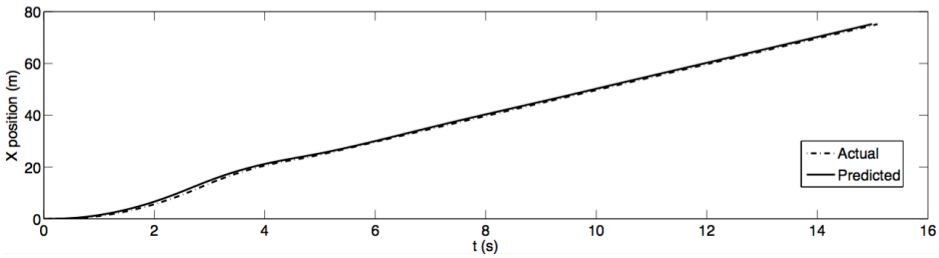


(a)

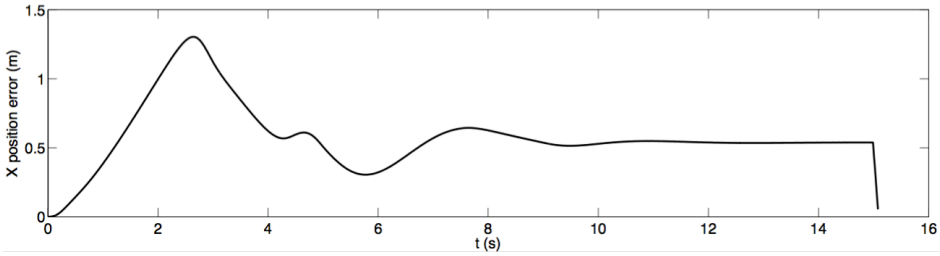


(b)

Figure 6. Spatial comparison (a) predicted and true trajectories (b) prediction errors



(a)

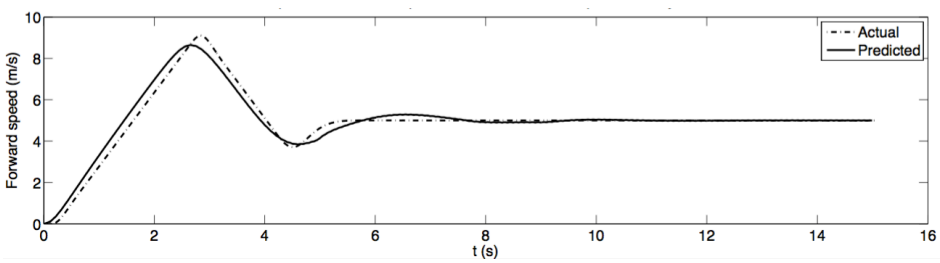


(b)

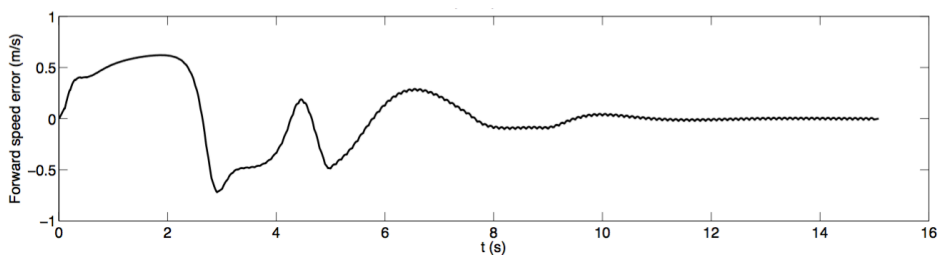
Figure 7. Temporal comparison of x positions (a) predicted and true trajectories (b) prediction errors

Figure 6(a) depicts the comparison between the true and predicted trajectories in the X-Y space. It is noted that predictions are pretty accurate in the spatial dimension. As shown in Fig. 6(b), the lateral errors are less than 0.25 meters over the entire trajectory. It is possible that the trajectory prediction may not converge to the truth as time proceeds because the error may accumulate and cause the prediction to diverge away eventually. The results from Fig. 6(a) also demonstrated that trajectory prediction errors didn't diverge although the training errors (especially for velocities) presented in previous section were not small. However, the condition of stability remains unclear, and needs to be studied in future work.

Both spatial and temporal precisions are required for an accurate 4D trajectory prediction, Fig. 7(a) shows the comparison of x position's time histories between the predicted and true trajectories. The x position errors were shown in Fig. 7(b)) and were bounded between 0 and 1.4 meters. This reveals that the predicted trajectory from the neural network is a bit fast and ahead of the true trajectory. Fig. 8(a) shows the comparison of forward speed's time histories between the predicted and true trajectories. The errors were shown in Fig. 8(b)) and were bounded between -0.7 and 0.6 meter per second. The comparison shows that, despite of the similar trend, the temporal difference between forward speeds is relatively high.



(a)



(b)

Figure 8. Temporal comparison of forward speeds (a) predicted and true trajectories (b) prediction errors

Table 2. Scenarios for trajectory prediction verification

Case	1a	1b	1c	2a	2b	2c	3a	3b	3c	4a	4b	4c
Vehicle ground speed (m/s)	2.0	2.0	2.0	5.0	5.0	5.0	8.0	8.0	8.0	11.0	11.0	11.0
Cross wind speed (m/s)	0.7	2.4	7.5	0.9	5.0	8.6	1.4	6.7	9.5	1.9	4.0	9.2

This neural network was then applied to predict trajectory in various scenarios, where the cross wind is in the range of $[0.7, 9.2]$ and the vehicle desired ground speed is within $[2.0 - 11.0]$. Table 2 lists the parameters for 12 cases used for verification and Fig. 9 plots the mean square errors and the maximimum absolute errors (MAE) between the predicted and actual trajectories. As discussed in previous sections, errors associated with the'X-Y' are spatial errors and errors associated with the 'T-X' and 'T-Y' are measured in temporal dimension. Spatial errors (both MSEs and MAEs) are usually smaller than the other errors and most of them

were bounded by ±0.5 meters, which means the predicted trajectories are aligned with actual trajectories well if the time was not taken into account. Temporal errors are usually a bit high, especially when the cross wind is strong. Cross track errors are typically less than along-track errors and, not surprisingly, all MAEs are higher than corresponding MSEs. From the figure, it is noted that all errors are well kept below 2 meters, which shows that the neural network approach has great promise in modeling sUAV's trajectories.

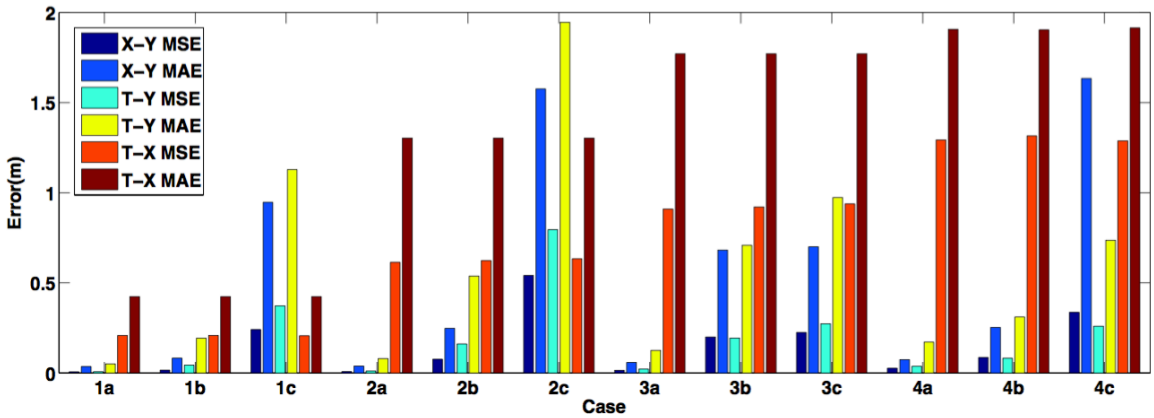


Figure 9. Errors between the predicted trajectory and the truth in various cases

IV. Conclusion

This work proposed a neural network approach for modeling sUAV's dynamics and control system for trajectory prediction. The motivation was to model sUAV's trajectory in a practical way such that the needs of intensive wind tunnel tests and flight tests are minimized while the fidelity of trajectory is not reduced much. If successful, this approach will also greatly reduce computational time when calculating trajectories. A neural network composed of nine inputs, 20 hidden neurons, and six outputs was proposed and trained using the back-propagation algorithm with the hybrid delta-bar-delta learning rule. As an initial work, a quadrotor vehicle model associated with a PD controller was used to generate trajectory data for training and testing purposes. Various trajectories were used to train the neural network with different vehicle ground speeds and side wind conditions. After the neural net is fully trained, it is used to predict future flight states at next time step based on current flight state information. This proposed approach was then verified with different but similar set of scenarios. The results show that the neural network can be trained to fit the train data. It can be used in a sequential manner to predict the trajectory step by step without diverging away from the truth. In addition, the 4-D trajectory prediction errors are less than 2.0 meters while the spatial errors are even less. Overall, the proposed approach shows great promise in modeling sUAV trajectories.

In the future work, further research/experiments are needed to improve this method. First, training data need to be much selective and representative to increase the efficiency and effectiveness of training. Second, thorough experiments are needed to prove its application with wide-range flight conditions and various vehicle types. Third, the conditions of stable trajectory prediction need to be studied and understood. Finally, different machine learning methods can be explored to identify good machine learning and training algorithms.

References

[1]Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., and Robinson, J. E., "Unmanned Aircraft System Traffic Management (UTM) Concept of Operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, Washington, D.C., 13-17 June 2016.

[2]Kopardekar, P., "Unmanned Aerial System Traffic Management (UTM): Enableing Low-altitude Airspace and UAS Operations," Tech. Rep. NASA TM-2014-218299, 2014.

[3]Lee, H. and Erzberger, H., "Time Controlled Descent Guidance Algorithm for Simulation of Advanced ATC Systems," Tech. Rep. Tech. Rep. NASA Technical Memorandum 84373, NASA Ames Research Center, Moffett Field, CA, August 1983.

[4]Erzberger, H. and Chapel, J., "Ground Based Concept for Time Control of Aircraft Entering the Terminal Area," *AIAA Guidance, Navigance, and Control Conference*, Snowmass, CO, 19-21 August 1985.

[5]Slattery, R. and Zhao, Y., "Trajectory Synthesis for Air Traffic Automation," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, 1997.

[6]Erzberger, H., Davis, T. J., and Green, S. M., "Design of Center-TRACON Automation System," *AGARD Meeting on Machine Intelligence in Air Traffic Management*, Berlin, Germany, 11-14 May 1993.

[7]Erzberger, H., Paielli, R., Isaacson, D., and Eshow, M. M., "Conflict Detection and Resolution In the Presence of Prediction Error," *1st USA/Europe Air Traffic Management Research and Development Seminar*, Saclay, France, 17-20 June 1997.

[8]Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," *3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy, 13-16 June 2000.

[9]Sweet, D. N., Manikonda, V., Aronson, J. S., Roth, K., and Blake, M., "Fast-time Simulation System For Analysis of Advanced Air Transportation Concepts," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Monterey, California, 5-8 August 2002.

[10]Kuchar, J. K. and Yang, L. C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions On Intelligent Transportation Systems*, Vol. 1, No. 4, 2000.

[11]Hwang, I. and Tomlin, C., "Protocol-based Conflict Resolution for Finite Information Horizon," *Proceedings of the American Control Conference*, Anchorage, AK, 8-10 May 2002.

[12]Farrell, J. A., Givargis, T. D., and Barth, M. J., "Real-Time Differential Carrier Phase GPS-Aided INS," *IEEE Transactions On Control Systems Technology*, Vol. 8, No. 4, 2000.

[13]Blomenhofer, H., Hein, G. W., Blomenhofer, E. T., and Werner, W., "Development of a Read-Time DGPS System in the Centimeter Range," *IEEE Position Location and Navigation Symposium*, Las Vegas, NV, 11-15 April 1994, pp. 532–539.

[14]Shim, D. H., Kim, H. J., and Sastry, S., "Control System Design for Rotorcraft-based Unmanned Aerial Vehicles using Time-domain System Identification," *Proceedings of the 2000 IEEE International Conference on Control Applications*, Anchorage, Alaska, 25-27 September 2000.

[15]Conroy, J. K., Humbert, J. S., and Pines, D. J., "System Identification of a Rotory-Wing Micro Air Vehicle," *Journal of the American Helicopter Society*, Vol. 56, No. 2, 2011.

[16]Labiod, S. and Guerra, T. M., "Fuzzy Adaptive Control For A Class Of Nonlinear Systems with Unknown Control Gain," *Evolving Systems*, Vol. 3, 2012.

[17]Dorobantu, A., Murch, A., Mettler, B., and Balas, G., "System Identification for Small, Low-Cost, Fixed-Wing Unmanned Aircraft," *AIAA Journal of Aircraft*, Vol. 50, No. 4, 2013.

[18]Beard, R. W., "Quadrotor Dynamics and Control," Tech. rep., October 2008.

[19]Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, 1989.

[20]Hornik, K., "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, Vol. 4, No. 2, 1991.

[21]Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, Vol. 2, No. 4, 1989.

[22]Grino, R., Cembrano, G., and Torras, C., "Nonlinear System Identification Using Additive Dynamic Neural Networks - Two Online Approaches," *IEEE Transactions On Circuits And Systems - Part I: Fundamental Theory And Applications*, Vol. 47, No. 2, 2000.

[23]Chen, S., Billings, S. A., and Grant, P. M., "Non-linear System Identification Using Neural Networks," *International Journal of Control*, Vol. 51, No. 6, 1990.

[24]Hu, Z. and Balakrishnan, S. N., "Parameter Estimation in Nonlinear Systems Using Hofield Neural Networks," *AIAA Journal of Aircraft*, Vol. 42, No. 1, 2005.

[25]Jacobs, R. A., "Increased Rates of Convergence Through Learning Rate Adaptation," *Neural Networks*, Vol. 1, 1988.

[26]Rumelhart, D. E. and McClelland, J. L., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge MA*, 1986.

[27]Bouabdallah, S. and Siegwart, R., "Full Control of a Quadrotor," *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct 29 - Nov 2 2007.

[28]Altug, E., Ostrowski, J. P., and Mahony, R., "Control of a Quadrotor Helicopter Using Visual Feedback," *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, Washington, D.C., May 2002.