

3-D Trajectory Modeling for Unmanned Aerial Vehicles

Baoqian Wang, Junfei Xie [†]
Texas A&M University-Corpus Christi

Yan Wan [‡]
University of Texas at Arlington

Gabriel Alexis Guizarro Reyes[§], Luis Rodolfo Garcia Carrillo [¶]
Texas A&M University-Corpus Christi

The burgeoning use of unmanned aerial vehicles (UAVs) evidences forthcoming environments where innumerable UAVs will appear in the National Airspace System (NAS). The UAS traffic management (UTM) aims to provide solutions to enable safe integration of numerous UAVs into the NAS, but the design of effective UTM strategies faces significant challenges. One of the challenges is to develop high-fidelity trajectory models for UAVs of partially known or unknown dynamics. Traditional physics-based models that require costly system identifications and field tests, and data-based models that require large amount of real flight data may not be feasible. To address this challenge, this paper introduces a hybrid 3-dimensional (3-D) UAV trajectory modeling framework, which integrates the physics-based and data-based models to capture the dynamics of UAVs of interest with high accuracy using only a small amount of real flight data. Simulation studies and field tests validate and demonstrate the good performance of the proposed framework.

I. Introduction

In recent years, small unmanned aerial vehicles (UAVs) have attracted significant interests from industry, academia, and government, featuring in a wide range of applications such as package delivery¹, agriculture spraying,² reconnaissance³, and search and rescue⁴. According to the U.S. Federal Aviation Administration (FAA), the number of hobbyist UAV purchases will grow from 1.9 million in 2016 to 4.3 million by 2022.⁵ Such burgeoning use of UAVs evidences forthcoming environments where innumerable UAVs will appear in the National Airspace System (NAS). However, how to integrate numerous UAVs into the NAS and how to manage the UAV traffic to ensure safe, efficient and flexible UAV operations is still an open problem. To address these challenges NASA starts developing prototype technologies for an UAV traffic management (UTM) system,^{6,7} which will provide airspace integration requirements to enable safe and efficient UAV operations in the civilian low-altitude airspace. Although past experiences on the management of airline traffic could be leveraged to build such a system, many research problems need to be addressed taking into account the considerable differences between manned civilian aircraft and small UAVs, including the high variety and uncertainty in UAV trajectories, heterogeneity of UAV types, and sensitivity to environmental disturbances. One of the major research problems in the UAV domain corresponds to trajectory modeling, which serves as the foundation for the evaluation and design of UTM strategies and is the focus of this study.

Collision avoidance among multiple UAVs and mitigation of traffic congestion can be achieved by altering UAVs' flight trajectories, which requires the prediction of UAVs' future movements. Manned or large unmanned aircraft are robust to uncertain environmental disturbances like winds, and their trajectory tracking

^{*} PhD student, Department of Computing Sciences, Texas A&M University-Corpus Christi.

[†] Assistant Professor, Department of Computing Sciences, Texas A&M University-Corpus Christi, AIAA member. Email: junfei.xie@tamucc.edu.

[‡] Associate Professor, Department of Electrical Engineering, University of Texas at Arlington, AIAA senior member.

[§] Master student, Department of Computing Sciences, Texas A&M University-Corpus Christi.

[¶] Assistant Professor, Department of Electrical Engineering, Texas A&M University-Corpus Christi. Email: luis.garcia@tamucc.edu.

errors are typically of little concern when planning their paths.⁸ However, small UAVs are sensitive to environmental disturbances and thus their trajectories may deviate from the desired ones significantly, causing relatively large trajectory errors that cannot be ignored. Therefore, to design effective UTM strategies, a high-fidelity trajectory model capable of predicting the response of UAVs to trajectory commands and environmental disturbances with high accuracy is required.

In disturbance-free environments (e.g., absence of wind), the trajectory tracking performance of small UAVs is dominated by their controllers, and different controllers typically have different trajectory tracking capabilities. Therefore, to build a high-fidelity trajectory model, both the dynamics of UAVs' plants and controllers need to be identified. With the advancement of UAV technology and increasing purchases of hobbyist UAVs, commercial UAVs usually come with a well-designed control system of automatic position tracking capability. However, the design details of the UAV system, including the controller, are usually kept confidential by UAV manufactures, making the trajectory modeling challenging. Existing studies on UAV modeling or trajectory tracking^{9–14} are mostly focused on the modeling for the UAV's plant, based on which high-performance controllers are designed to track a desired target. In most of these studies, physically-based models are used to describe the dynamics of UAVs' plants, where model parameters are derived through system identification. However, to simplify the controller design, linearized or simplified mathematical equations are usually used to approximate the complicated dynamics of UAVs under assumptions such as rigidity and symmetry,^{15–17} introducing unavoidable model errors.

Considering the high costs of system identification and the requirement for knowledge of systems' physical dynamics, data-based models like neural networks (NN) have recently attracted significant interest. These data-based models are able to approximate the dynamics of UAVs¹⁸ using data only and do not require explicit knowledge of the underlying physical laws. Based on these models, various control approaches,^{19–21} such as adaptive NN based control²² and reinforcement learning,²³ are then developed. However, to obtain a high-fidelity data-based model, large amount of data is required. Although UAV plant modeling can be addressed using the aforementioned existing approaches, how to model the trajectory of a UAV with unknown onboard controllers is still challenging. In the manned aviation, trajectory models can be built with sufficient accuracy without knowing the control systems of the aircraft, if the aerodynamics are known.² However, the great effort and high cost required to obtain the aerodynamics, which require large amount of wind tunnel and flight tests, make this approach unfeasible for low-cost small UAVs. Reference² introduces a simple NN-based trajectory model that regards the whole UAV system including the plant and the onboard controller as a black-box. However, it does not consider the altitude control. Systematic investigation on the accuracy and robustness of the model under wind disturbances is also lacking.

In this paper, we introduce a hybrid 3-dimensional (3-D) UAV trajectory modeling framework, which integrates the physically-based and data-based models. The key idea is to use a physically-based model, which may not perfectly capture the true dynamics of the UAV of interest, to generate large amount of trajectory data and use these data to train a data-based model. This baseline model is then tuned using small amount of real flight data to capture the true dynamics of the targeted UAV. This hybrid 3-D trajectory modeling framework is promising in that 1) it only requires a small amount of real flight data, saving significant amount of costs for field tests; 2) it requires only one intensive model training; 3) the trajectory model can be quickly deployed for different types of UAVs; and 4) it addresses the limitations of physically-based and data-based models, while retaining their advantages.

In the rest of the paper, we first review a physically-based model in Section II to describe the dynamics of quadrotor UAVs with proportional-derivative (PD) controllers under wind disturbances. Two data-based models are then described in Section III. In Section IV, we conduct simulation studies and field tests to validate the proposed framework and evaluate its performance. Section V finally concludes the paper.

II. A Physically-based Model for Quadrotor UAVs

In this section, we briefly introduce a physically-based dynamic model for quadrotor UAVs stabilized by means of PD controllers, and affected by wind disturbances (see Figure 1). Quadrotor UAVs are selected in this work, as they are widely used in commercial applications due to their high maneuverability, as well as vertical take-off and landing capabilities.¹⁵ PD controllers are selected as they are one of the most commonly adopted controllers.²⁴

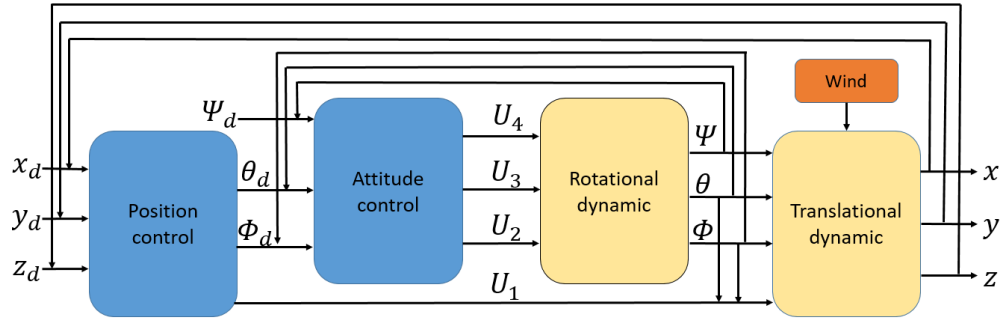


Figure 1: Block diagram of a physically-based model for quadrotor UAVs with PD controllers.

II.A. Plant Model for Quadrotor UAVs

The typical plant model for quadrotor UAVs contains two sub-systems: one captures the rotational dynamics and the other one captures the translational dynamics describing the UAV motion in the 3-D space. By neglecting gyroscopic effects and the moment of inertia effects³, the rotational dynamics of quadrotor UAVs can be described by the following equation

$$\begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{U_2}{I_{bx}} \\ \frac{U_3}{I_{by}} \\ \frac{U_4}{I_{bz}} \end{bmatrix}$$

where I_{bx} , I_{by} , I_{bz} are the moments of inertia along the principal axes in the body frame φ , θ , ψ represent the roll, pitch, and yaw angles in the body frame, respectively. U_2 , U_3 , and U_4 are the control inputs. Under the assumptions that the UAV's structure is rigid and symmetrical,¹³ the translational dynamics that consider wind disturbances can be captured using Euler-Lagrange equations as follows

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} -(\cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \theta \sin \psi) \frac{U_1}{m} \\ (-\cos \varphi \sin \theta \sin \psi + \sin \varphi \cos \theta) \frac{U_1}{m} \\ -g + \cos \varphi \cos \theta \frac{U_1}{m} \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x + w_x \\ v_y + w_y \\ v_z + w_z \end{bmatrix} \quad (1)$$

where (x, y, z) represent the 3-D coordinates of the UAV in the inertial frame. g is the gravitational acceleration, and U_1 is the control input. (v_x, v_y, v_z) and (w_x, w_y, w_z) represent the translational velocity of the UAV and the wind velocity in the inertial frame, respectively.

II.B. PD Control System

To generate the control inputs U_i , $i \in \{1, 2, 3, 4\}$, a PD-based control system¹³ is used. This control system consists of two sub-systems, one for position control and the other one for attitude control. The PD position control sub-system can be described by the following equation

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

where k_p and k_d denote the proportional and derivative gains, respectively. (x_d, y_d, z_d) , $(\dot{x}_d, \dot{y}_d, \dot{z}_d)$ and $(\ddot{x}_d, \ddot{y}_d, \ddot{z}_d)$ represent the desired position, desired velocity and desired acceleration, respectively. The outputs (i.e., $(\ddot{x}_d, \ddot{y}_d, \ddot{z}_d)$) of the position control sub-system are the inputs to the attitude control sub-system

described by following equations

$$\begin{aligned} \frac{U_1}{m} &= \frac{z_d + g}{\cos \varphi \cos \theta} \\ \varphi_d &= \frac{m}{U_1} \begin{bmatrix} -\sin \psi & -\cos \psi & x_d \\ \cos \psi & -\sin \psi & y_d \end{bmatrix} \\ \theta_d &= \frac{1}{U_1} \begin{bmatrix} k_{p,\varphi}(\varphi_d - \varphi) + k_{d,\varphi}(\dot{\varphi}_d - \dot{\varphi}) \\ k_{p,\theta}(\theta_d - \theta) + k_{d,\theta}(\dot{\theta}_d - \dot{\theta}) \end{bmatrix} L \\ U_4 &= k_{p,\psi}(\psi_d - \psi) + k_{d,\psi}(\dot{\psi}_d - \dot{\psi}) \end{aligned}$$

where $k_{p,\varphi}$, $k_{d,\varphi}$, $k_{p,\theta}$, $k_{d,\theta}$, $k_{p,\psi}$ and $k_{d,\psi}$ are the gains of PD controllers. φ_d , θ_d , ψ_d represent the desired roll, pitch, and yaw angles, respectively. L is the distance between rotors and the center of gravity. The outputs of the position and attitude control sub-systems, including U_1 , U_2 , U_3 , U_4 , are the control inputs to the plant model of the quadrotor UAVs. The complete physically-based model that combines both plant and control models for quadrotor UAVs is shown in Figure 1.

II.C. Wind Disturbance Model

In this study, we adopt the Dryden wind gust model^{25, 26} to generate random wind velocities (w_x , w_y , w_z) in Equation (1). This model contains two components: a static component that remains constant during the flight of the UAV, and a non-static component that varies over time. The wind disturbance value $w(t)$, $w \in \{w_x, w_y, w_z\}$, can be generated using the following equation

$$w(t) = w_s + w_n(t)$$

where w_s is a constant and is obtained through meteorological observation.^{25, 26} $w_n(t)$ varies over time and is obtained by summing a set of sinusoidal excitations as follows

$$w_n(t) = \sum_{i=1}^n a_i \sin(\Omega_i t + \gamma_i)$$

where n is the total number of sinusoidal functions. Ω_i and γ_i are parameters to be configured. a_i is the magnitude given by

$$a_i = \sqrt{\frac{p}{\Delta\Omega_i \Phi(\Omega_i)}}$$

where $\Delta\Omega_i = \Omega_{i+1} - \Omega_i$. $\Phi(\Omega_i)$ is the power spectral density calculated by following equations

$$\begin{aligned} \Phi(\Omega) &= \begin{cases} \Phi_h(\Omega), & \text{If } w \in \{w_x, w_y\} \\ \Phi_v(\Omega), & \text{If } w = w_z \end{cases} \\ \Phi_h(\Omega) &= \sigma_h^2 \frac{2L_h}{\pi} \frac{1}{1 + (L_h\Omega)^2} \\ \Phi_v(\Omega) &= \sigma_v^2 \frac{2L_v}{\pi} \frac{1 + 3(L_v\Omega)^2}{[1 + (L_v\Omega)^2]^2} \\ L_h &= \frac{L_v}{(0.177 + 0.000823z)^2} \\ L_v &= |z| \\ \sigma_h &= \frac{\sigma_v}{(0.177 + 0.000823z)^4} \\ \sigma_v &= 0.1\xi \end{aligned}$$

where $\Phi_h(\Omega)$ and $\Phi_v(\Omega)$ are power spectral densities along the horizontal and vertical dimensions, respectively. L_h and L_v are the length scales along the horizontal and vertical dimensions, respectively. σ_h and σ_v are the turbulence intensities along the horizontal and vertical dimensions, respectively. z is the altitude of the UAV in the inertial frame. ξ denotes the wind speed in knots at 20 ft altitude.

III. Neural Network-based 3-D Trajectory Models for Quadrotor UAVs

In order to capture the 3-D trajectory patterns for quadrotor UAVs, in this section, we introduce two different NN-based models. The first one corresponds to a standard NN, while the second one corresponds to a more advanced NN called long short-term memory (LSTM).

III.A. Standard NN-based 3-D Trajectory Model

NN models²⁷ have been heavily explored in various fields, due to its simple structure and the capability to learn complex nonlinear system dynamics without knowing the inherent physical laws. In this work, we first build a simple standard NN-based 3-D trajectory model, by regarding the whole UAV system as a black-box with unknown dynamics, and designing a NN model to capture the relationship between the inputs and output of this black-box. In particular, the inputs to the UAV system (see Figure 1) include the desired position, $(x_d(t), y_d(t), z_d(t))$, desired yaw angle, ψ_d , and environmental disturbances captured by $(w_x(t), w_y(t), w_z(t))$. As feedback controls are typically used for UAV trajectory tracking, UAV's current position at time t , $(x(t), y(t), z(t))$, should also be the input. The output of the UAV system is the position of UAV at time $t + 1$. An illustration of the designed standard NN-based 3-D trajectory model is shown in Figure 2, which consists of three layers with 10 input units, 10 hidden units and 3 output units. Note that many UAV systems have built-in sensors that can measure system states including velocity, acceleration, roll, pitch and yaw angles. These additional information, if included as inputs to the NN-based model, may help improve the model accuracy, but are not needed in this study which focuses on UAVs' trajectories. Besides, introducing additional inputs increases the amount of data required for training.

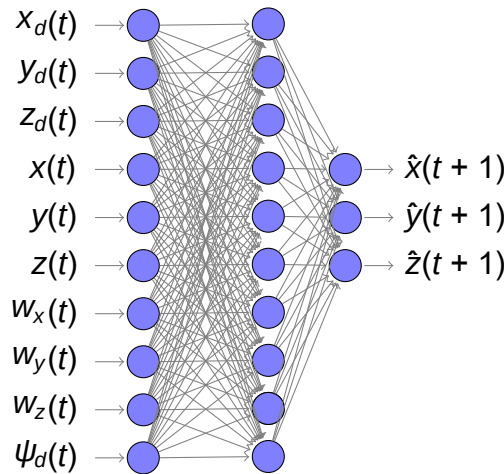


Figure 2: Illustration of the proposed standard NN-based 3-D trajectory model for quadrotor UAVs.

Let us now describe the mathematical formulation of this standard NN-based 3-D trajectory model. For ease of presentation, we let $X(t) = [x_d(t), y_d(t), z_d(t), x(t), y(t), z(t), w_x(t), w_y(t), w_z(t), \psi_d(t)]^0$ and $\hat{Y}(t+1) = [\hat{x}(t+1), \hat{y}(t+1), \hat{z}(t+1)]^0$ as the input and output vectors of the NN-based model, respectively, where 0 is the transposition operator. The relationship between the input and output vectors is described by the following equation

$$\hat{Y}(t+1) = W^{(2)} h(W^{(1)} X(t) + b_1) + b_2 \quad (2)$$

where $W^{(j)}$ is a weight vector from the j -th layer to the $(j+1)$ -th layer, $j \in \{1, 2\}$. $f(\cdot)$ is the rectified linear unit activation function,²⁸ and b_j is the bias term for the j -th layer.

To obtain the weight vectors $W^{(j)}$, we use the back propagation algorithm²⁹ and Adam optimizer³⁰ to estimate their values iteratively. In particular, we divide the training data into multiple batches $\{B_1, B_2, \dots, B_p\}$, where $p = \frac{N}{|B_k|}$ is the total number of batches, N is the total number of training examples, and $B = |B_k|$, $k = \{1, 2, \dots, p\}$, is the batch size. For each batch B_k , the weight vectors are updated using the following

equations

$$\begin{aligned}
 W_{k+1}^{(j)} &= W_k^{(j)} - \frac{\eta}{v_k + 1} m_k \\
 m_k &= \frac{m_k}{1 - \beta_1} \\
 v_k &= \frac{v_k}{1 - \beta_2} \\
 m_k &= \beta_1 m_{k-1} + (1 - \beta_1) h_k \\
 v_k &= \beta_2 v_{k-1} + (1 - \beta_2) h_k^2 \\
 h_k &= O_{W_k^{(j)}} J(W_k^{(j)}) \\
 J(W_k^{(j)}) &= \frac{1}{B} \sum_{t \in B_k} (Y(t) - \hat{Y}(t))^2
 \end{aligned}$$

where η , β_1 and β_2 are parameters to be configured. This training process is repeated for e epochs, where one epoch refers to one pass over all training examples.

III.B. LSTM-based 3-D Trajectory Model

The LSTM is a special type of recurrent NN that can learn time-dependent information in time series data³¹. It is composed of LSTM cells and a fully connected network as illustrated in Figure 3. To capture UAV movements, the LSTM-based 3-D trajectory model has input and output vectors taking the same formula as the ones adopted in the standard NN-based model, but utilizes more historical data for prediction. In particular, the UAV position at time $t + 1$, $\hat{Y}(t + 1)$, is predicted using data from previous T time steps, i.e., $\{X(t - T + 1), X(t - T + 2), \dots, X(t)\}$, where T controls the amount of historical data used for prediction.

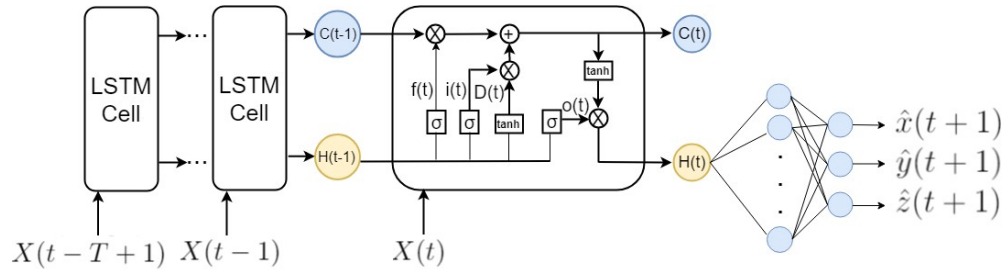


Figure 3: Architecture of the LSTM based 3-D trajectory model for quadrotor UAVs.

To predict $\hat{Y}(t + 1)$, the input vectors first pass through T LSTM cells, where each LSTM cell takes an input vector $X(t - l)$, $l \in \{0, 1, \dots, T - 1\}$, and calculates the cell state $C(t - l)$ and hidden state $H(t - l)$ using the following equations

$$\begin{aligned}
 C(t - l) &= f(t - l) \circ C(t - l - 1) + i(t - l) \circ D(t - l) \\
 H(t - l) &= o(t - l) \circ \tanh(C(t - l)) \\
 f(t - l) &= \sigma(W_f [H(t - l - 1), X(t - l)] + b_f) \\
 i(t - l) &= \sigma(W_i [H(t - l - 1), X(t - l)] + b_i) \\
 D(t - l) &= \tanh(W_D [H(t - l - 1), X(t - l)] + b_D) \\
 o(t - l) &= \sigma(W_o [H(t - l - 1), X(t - l)] + b_o)
 \end{aligned}$$

$f(t - l)$, $i(t - l)$, $D(t - l)$, $o(t - l)$ are the inner states of the LSTM cell at time $t - l$. W_f , W_i , W_D , W_o are the weights to be trained and b_f , b_i , b_D , b_o are biases. $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid function³² and the hyperbolic function,³³ respectively. $H(t - T) = C(t - T) = 0$. The operator $\circ$ is the Hadamard product (element-wise product).

The hidden state $H(t)$ is then passed to a fully connected network to calculate $\hat{Y}(t + 1)$ using the following equation

$$\hat{Y}(t + 1) = W_y(H(t) + b_y)$$

where W_y and b_y are weight and bias, respectively. To estimate the weights, the LSTM adopts a similar training procedure as the standard NN. The interested reader is referred to reference³¹ for a more detailed description of the training procedures.

IV. Numerical and Experimental Studies

IV.A. Simulation Studies

In this section, we implement the physically-based model introduced in Section II to generate trajectory data and use these data to train the data-based models described in Section III.A. Performance of the trained standard NN- and LSTM-based 3-D trajectory models are then evaluated.

IV.A.1. Trajectory Data Generation and Model Training

Trajectories are generated by configuring the desired position (x_d, y_d, z_d) , desired yaw angle ψ , and wind velocity (w_x, w_y, w_z) in the physically-based model described in Section II. Here we adopt the 3-D smooth-turn random mobility model developed in our previous studies^{34, 35} to generate the desired positions (x_d, y_d, z_d) . The desired yaw angle is set to $\psi_d = 0.5$. Wind velocities (w_x, w_y, w_z) are generated using the wind disturbance model described in Section II.C. In particular, we set $\xi = 2m/s$, $n = 5$, and let the static component of the wind velocities, w_s , randomly take a value from set $\{0m/s, 1m/s, \dots, 9m/s\}$ to capture different wind intensities. Ω_i and γ_i take values randomly from the ranges of $[0.1rad/s, 1.5rad/s]$ and $[-0.5rad, 0.5rad]$, respectively. The physically-based model is configured according to Table 1, with the sampling rate of the PD controllers set to $\Delta t = 0.01s$. By running this model, we generate 100 UAV trajectories with each lasting for 10s. Among these trajectories, one is randomly picked for testing and evaluation, with the others used for training. Figure 4(a) plots the UAV trajectory selected for testing. The associated wind velocities are shown in Figure 4(c), which are generated by setting w_s to $0m/s$, $2m/s$, and $4m/s$ within time intervals $[0s, 3s)$, $[3s, 6s)$, and $[6s, 10s)$, respectively. As we can see from Figure 4(a), the UAV is able to track the desired trajectory well under light winds, but the tracking performance degrades with the increase of wind speed. The corresponding trajectory tracking errors calculated by the following equation are shown in Figure 4(b),

$$\text{Error} = \|Y(t) - \hat{Y}(t)\| \quad (3)$$

where $\|\cdot\|$ is the euclidean norm operator.

Table 1: Configuration of the Parameters in the Physically-based Model for Quadrotor UAVs.

Parameters of the Plant Model		Parameters of the PD Control System	
g	$9.8m/s^2$	k_p	12.88
m	$1.2 kg$	k_d	12.88
L	$0.24 m$	$k_{p,\varphi}$	0.1
I_{bx}	$0.0085 kg \cdot m^2$	$k_{d,\varphi}$	0.2
I_{by}	$0.0085 kg \cdot m^2$	$k_{p,\theta}$	0.4
I_{bz}	$0.0158 kg \cdot m^2$	$k_{d,\theta}$	0.4
		$k_{p,\psi}$	0.048
		$k_{d,\psi}$	0.024

To train the standard NN- and LSTM-based trajectory models, the trajectory data are first pre-processed, with each value scaled to the range of $[0, 1]$ using the following equation

$$\hat{u}(t) = \frac{u(t) - u_{\min}}{u_{\max} - u_{\min}} \quad (4)$$

where $u \in \{x_d, y_d, z_d, x, y, z, w_x, w_y, w_z, \psi_d\}$, $u_{\min} = \min_t u(t)$ and $u_{\max} = \max_t u(t)$. These scaled data are then fed into the trajectory models to tune their parameters. The number of epochs and the batch size are set to $e = 8$ and $B = 5$, respectively. The parameters of the Adam optimizer are set to $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

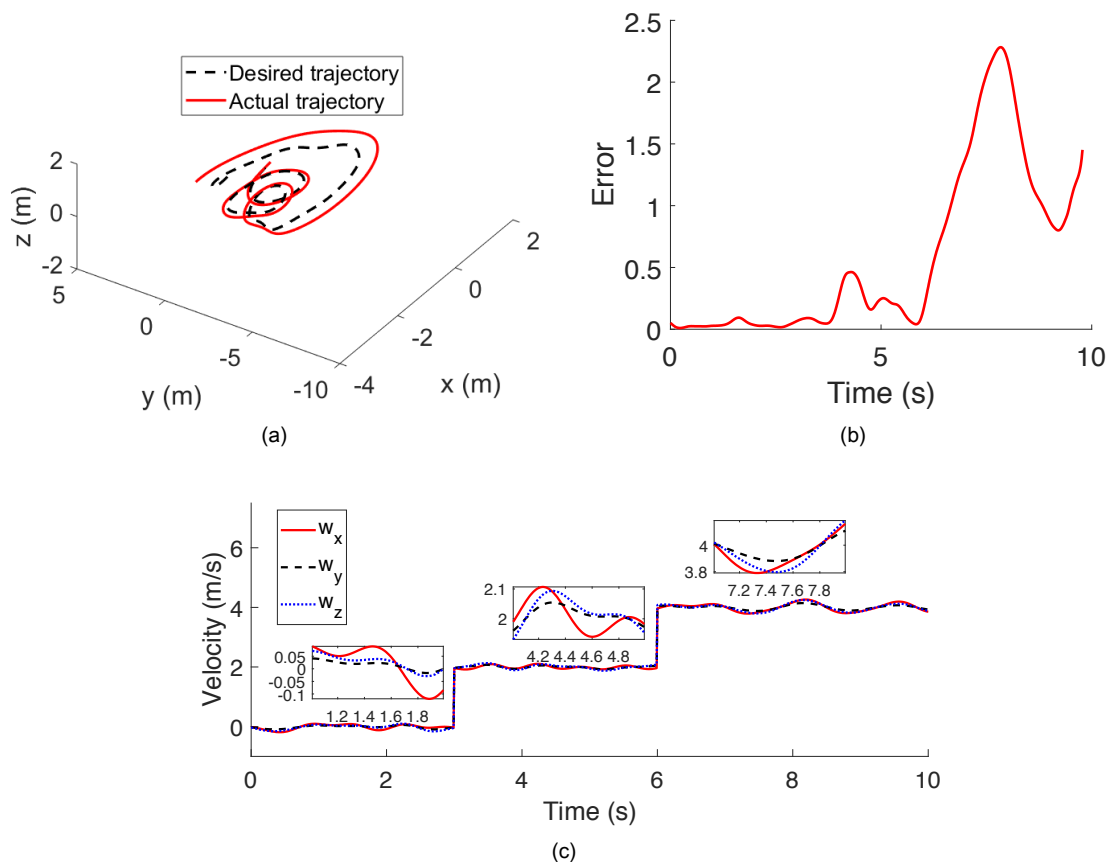


Figure 4: Illustration of the a) UAV trajectory used for testing; b) corresponding trajectory tracking errors; and c) wind velocities.

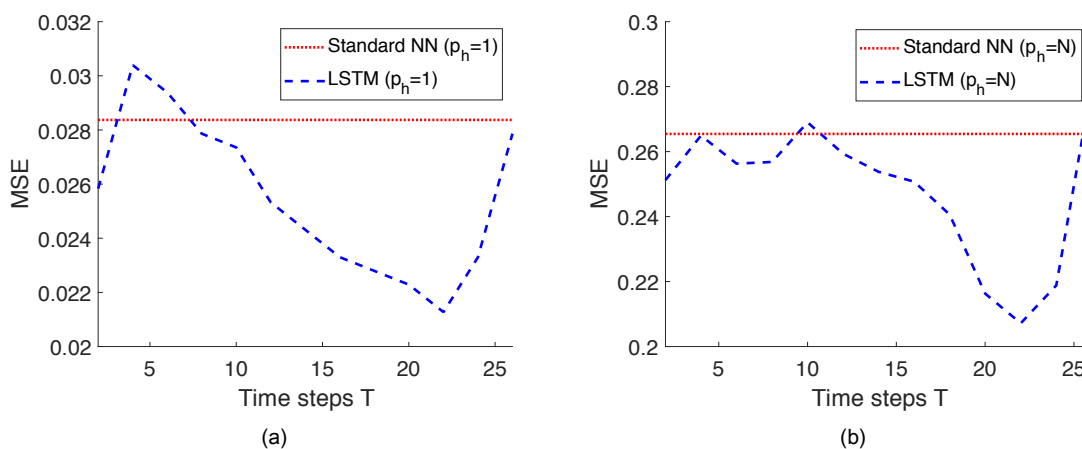


Figure 5: Prediction performance of the LSTM-based 3-D trajectory model, compared with the standard NN-based model, at different values of T in cases when the prediction horizon a) $p_h = 1$ and b) $p_h = N$.

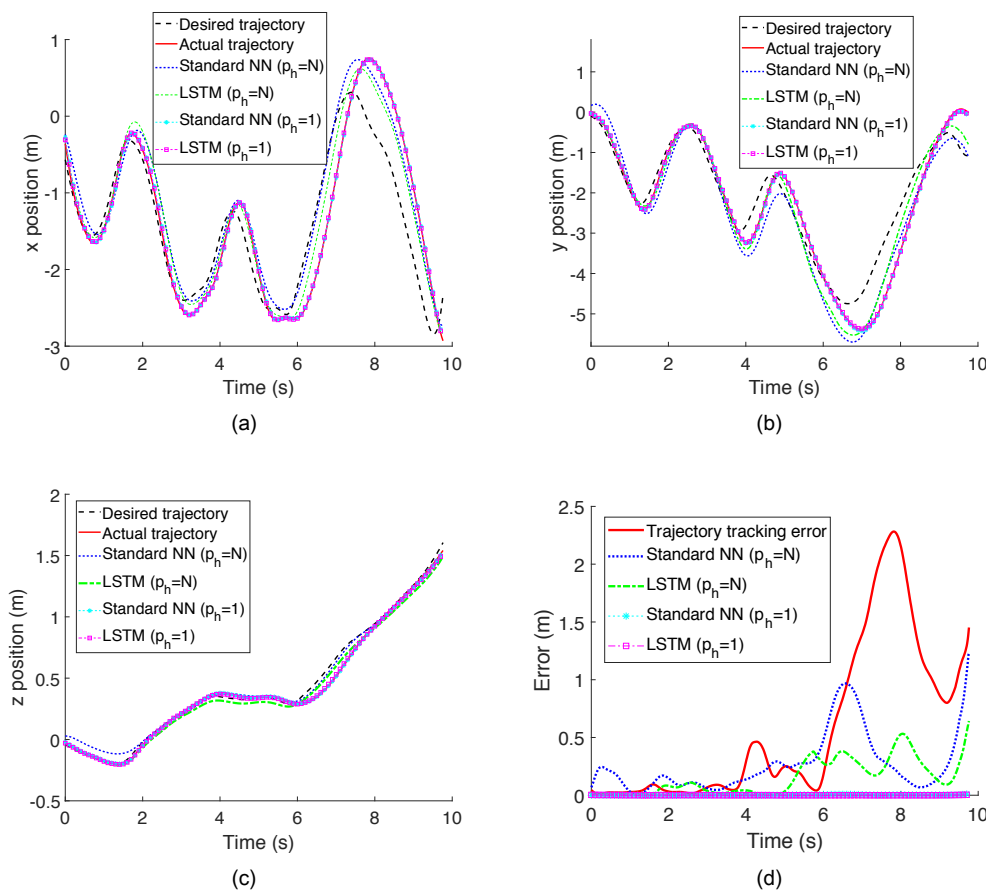


Figure 6: Trajectories along a) x, b) y, and 3) z directions predicted by the two NN-based models in different cases compared with the desired and actual trajectories. d) The corresponding prediction errors of the two NN-based models compared with the trajectory tracking errors.

IV.A.2. Performance Evaluation of the NN-based 3-D Trajectory Models

Two cases are evaluated in order to understand the capabilities of the two NN-based 3-D trajectory models. In the first case, we only feed the models with the initial UAV position and the desired trajectory. In this case, UAV position $\hat{Y}(t)$ predicted at time t is used to predict position $\hat{Y}(t+1)$ at the next time step, and the prediction horizon is $p_h = N$, where N is the total number of time steps. In the second case, we feed the models with the real UAV position $X(t)$ at each time step to predict the position $\hat{Y}(t+1)$ at the next time step, i.e., $p_h = 1$.

We first conduct an experiment to determine the optimal value of T in the LSTM-based model. Figure 5 shows the prediction performance of the LSTM-based model evaluated at different values of T in different cases, compared with the standard NN-based model, where mean squared error (MSE) is used to measure the prediction performance. As shown in Figure 5, the optimal prediction performance is achieved at $T = 22$, which is thus adopted in following studies.

With the optimal T in the LSTM-based model determined, we then conduct experiments to evaluate the performance of the two NN-based models. The UAV trajectories predicted by the two NN-based models in different cases, compared with the actual and desired trajectories, are shown in Figures 6(a)-6(c). The corresponding prediction errors, compared with the trajectory tracking errors, are presented in Figure 6(d). As expected, the performance of the two NN-based models degrades with the increase of the prediction horizon, due to accumulated errors. However, in both cases, the prediction errors of the two NN-based models are much smaller than the trajectory tracking errors most of the time, no matter how large the wind speed is. This indicates the necessity to develop trajectory models for UAVs of partially known or unknown



Figure 7: The Bebop quadrotor used in the flight tests.

dynamics, and also demonstrates the effectiveness of the proposed NN-based models. Comparing the two NN-based models, we can observe that the LSTM-based model outperforms the standard NN-based model in both cases, as it utilizes more historical information for prediction. In addition, the LSTM-based model is also more robust than the standard NN-based model to wind disturbances.

IV.B. Field Tests

Field tests were conducted making use of the Bebop quadrotor developed by Parrot, see Figure 7, which implements PD controllers to achieve position tracking. This multirotor platform was allowed to validate the proposed hybrid trajectory modeling framework, which first uses large amount of data generated from a physically-based model to train a data-based model and then uses small amount of real flight data to fine tune model parameters for higher accuracy. Two trajectory datasets were collected in an indoor environment, see Figure 8. The desired positions for the first trajectory are generated by the following equations

$$\begin{aligned}x_d(t) &= \rho(t) \cos(\lambda t) \\y_d(t) &= \rho(t) \sin(\lambda t) \\z_d(t) &= 1.2 \\ \rho(t) &= r \cos(\alpha t)\end{aligned}$$

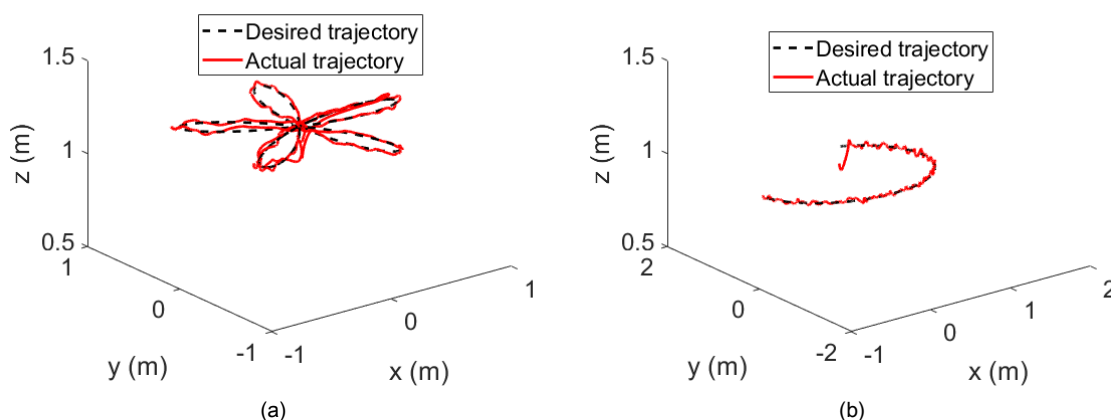


Figure 8: Two trajectories collected in the field tests using the Bebop quadrotor.

where $\lambda = 0.041$, $\alpha = 5$, $r = 0.8$. The equations to generate the desired positions for the second trajectory are given by

$$\begin{aligned}x_d(t) &= \sin(0.04t + 0.5) \\y_d(t) &= \cos(0.04t + 0.6) \\z_d(t) &= 1\end{aligned}$$

In both trajectories, the desired yaw angle is set to $\psi_d = 0$. Each trajectory lasts for 100s with a sampling rate of 0.01s. The position of the UAV was measured by means of a Vicon motion capture system.

To demonstrate the effectiveness of the proposed hybrid trajectory modeling framework, we use the first trajectory to further tune the parameters in the NN-based models obtained in Section IV.A. We then use the second trajectory for testing and performance evaluation. For comparison, we also evaluate the performance of the 1) NN-based models trained only using data generated from the physically-based model and 2) NN-based models trained only using the first trajectory. The results are shown in Figures 9-11. As the field tests are executed in an indoor environment, wind disturbances are assumed to be nonexistent. Therefore, the UAV is able to track the desired positions well. This makes the prediction errors of the proposed hybrid models shown in Figure 9 very close to the trajectory tracking errors. Comparing Figure 9 with Figure 10, we can observe significant performance improvement after fine tuning the parameters of the NN-based models using the real flight data. The NN-based models trained only using the first trajectory shown in Figure 11 has the worst performance, due to the lack of training data.

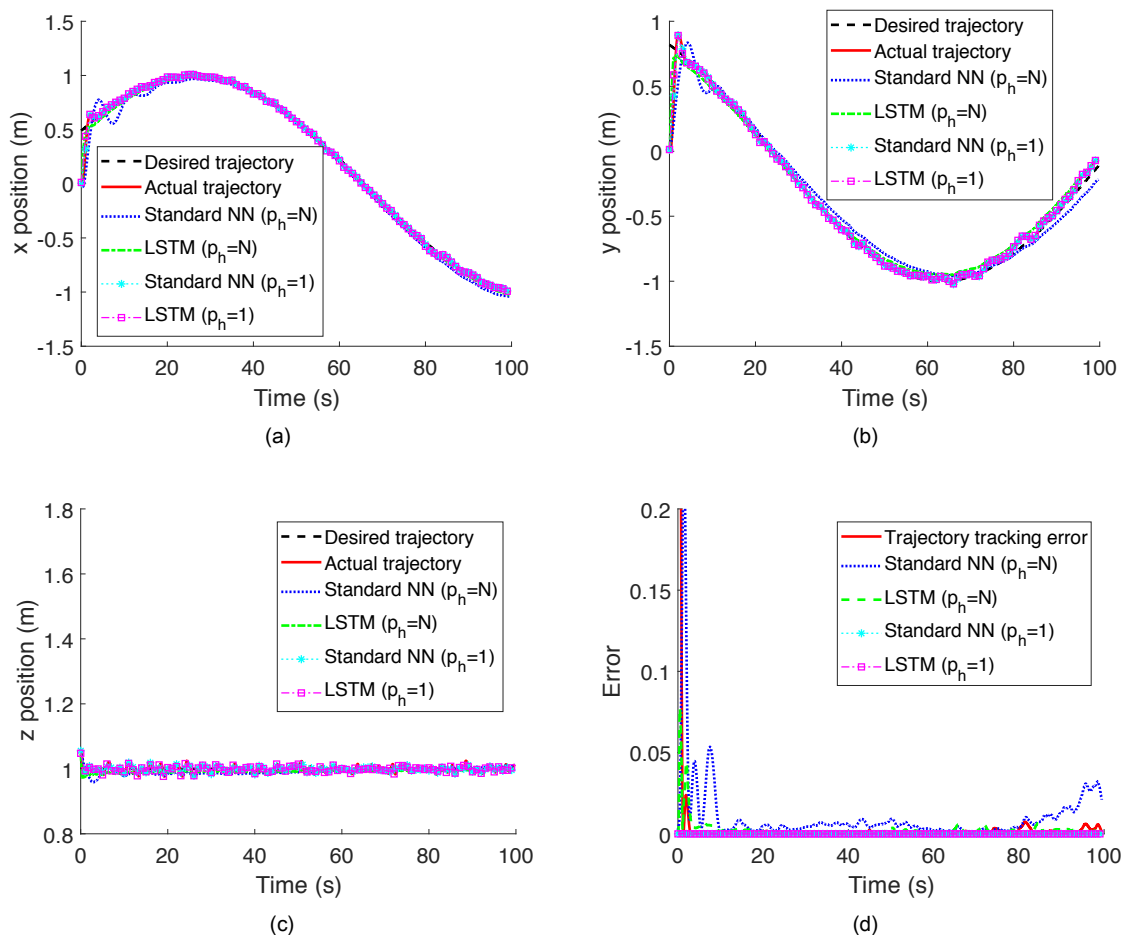


Figure 9: Performance of the two hybrid 3-D trajectory models in predicting the movements along the a) x, b) y and c) z directions of the Bebop quadrotor that follows the second trajectory. d) The corresponding prediction errors.

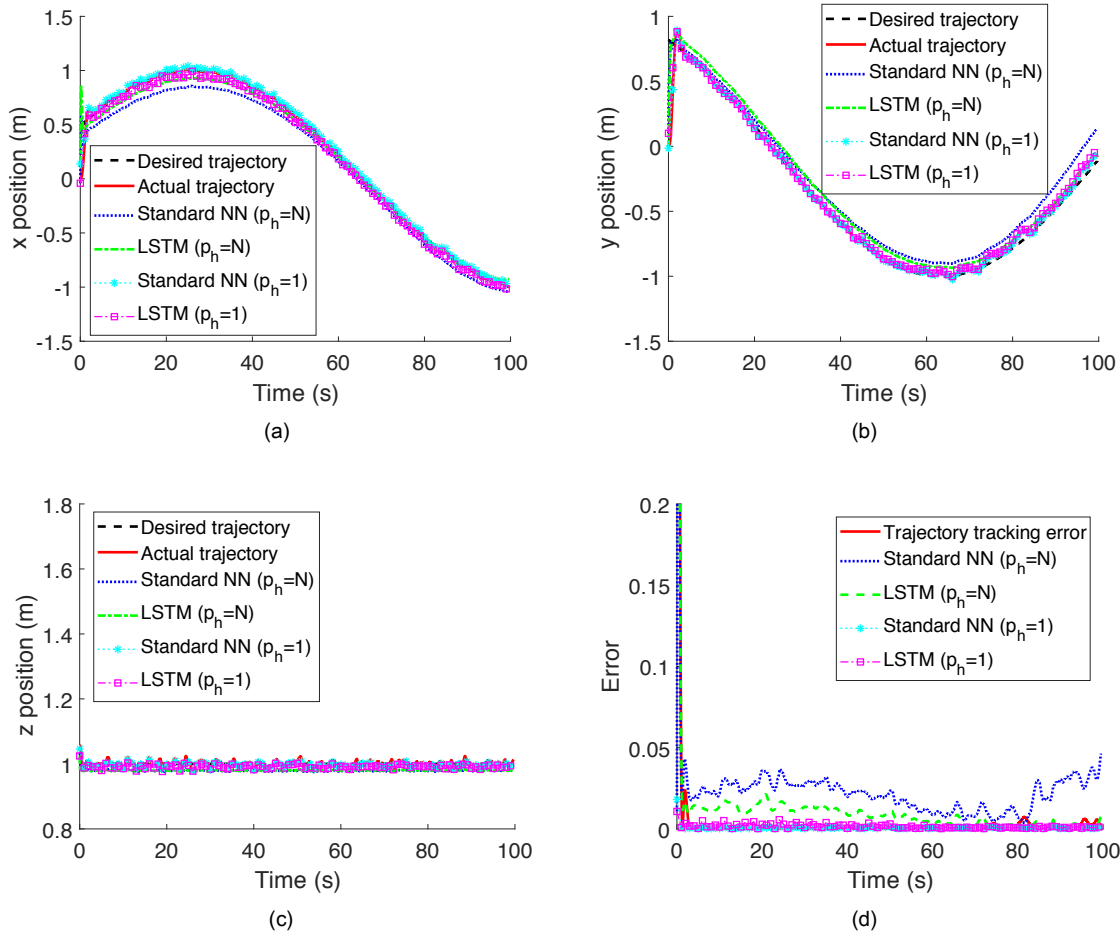


Figure 10: Performance of the two NN-based 3-D trajectory models trained only using data generated from the physically-based model in predicting the movements along the a) x, b) y and c) z directions of the Bebop UAV that follows the second trajectory. d) The corresponding prediction errors.

V. Conclusion

In this paper, the 3-D trajectory modeling problem for UAVs whose dynamics are partially known or unknown is investigated. This topic represents one of the most challenging research problems to be solved in order to design effective UTM strategies. Instead of developing a high-fidelity physically-based model through costly system identification and field tests, or training a data-based model using massive real flight data, we combined the advantages of both types of models. In particular, we developed a hybrid 3-D UAV trajectory modeling framework, which first uses data generated from a physically-based model of (possibly) low fidelity to train a data-based model, and then uses real flight data available to fine tune model parameters to capture the dynamics of the UAV of interest. The results of the simulation studies and field tests demonstrate the effective performance of the proposed modeling framework, which only requires a small amount of real flight data to achieve high accuracy.

Acknowledgments

This work is partially supported by the National Science Foundation under Grant 1730675, 1730589, 1839707 and 1839804 by Department of Defense, Network Science Division under grant W911NF1810210, and by the Texas A&M University-Corpus Christi Research Enhancement program.

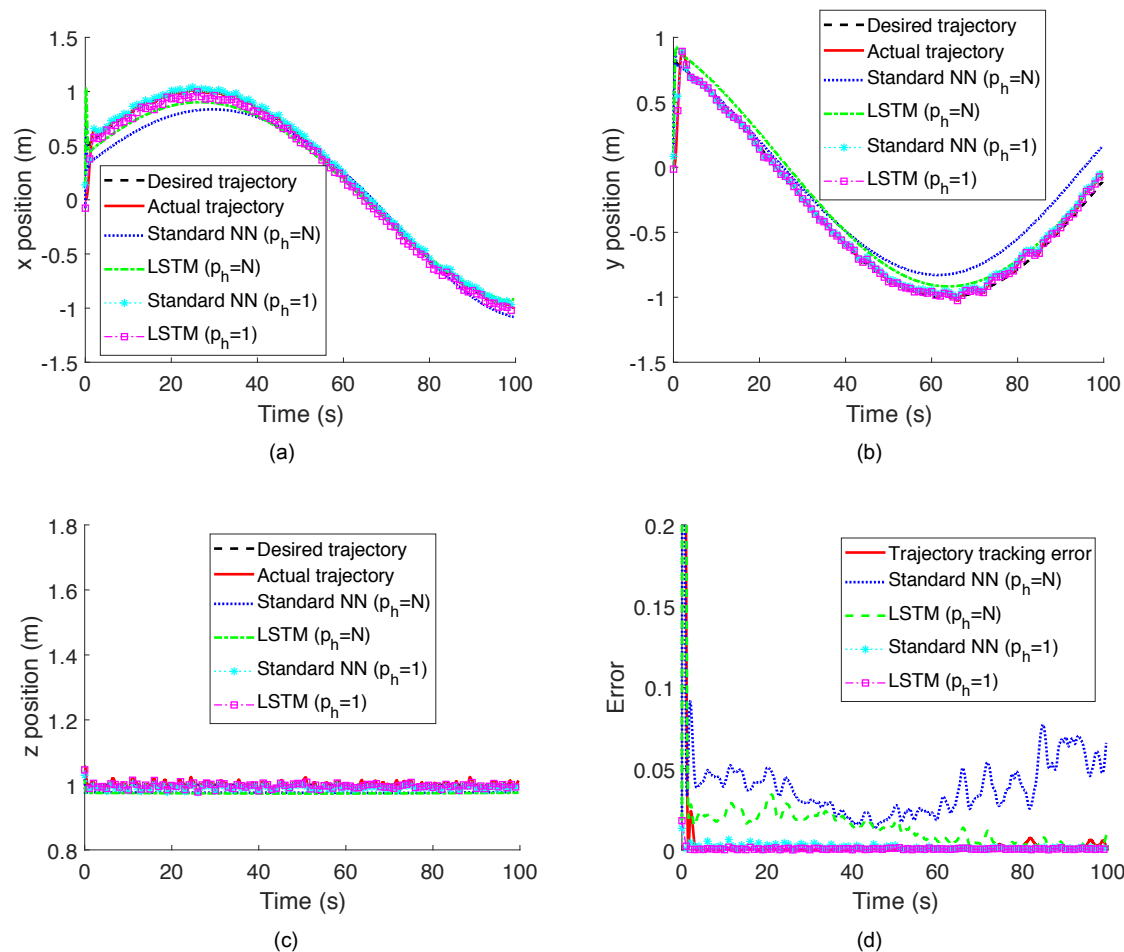


Figure 11: Performance of the two NN-based 3-D trajectory models trained only using the first trajectory in predicting the movements along the a) x, b) y and c) z directions of the Bebop quadrotor that follows the second trajectory. d) The corresponding prediction errors.

References

- ¹J. Stolaroff, "The need for a life cycle assessment of drone-based commercial package delivery," Lawrence Livermore National Laboratory (LLNL), Livermore, CA, Tech. Rep., 2014.
- ²Y. Huang, W. C. Hoffmann, Y. Lan, W. Wu, and B. K. Fritz, "Development of a spray system for an unmanned aerial vehicle platform," *Applied Engineering in Agriculture*, vol. 25, no. 6, pp. 803–809, 2009.
- ³M. Bhaskaranand and J. D. Gibson, "Low-complexity video encoding for uav reconnaissance and surveillance," in *Military Communications Conference*. IEEE, 2011, pp. 1633–1638.
- ⁴P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2007, pp. 1–13.
- ⁵"FAA Releases 2016 to 2036 Aerospace Forecast," Accessed: June 1, 2018. [Online]. Available: <https://www.faa.gov/news/updates/?newsId=85227>
- ⁶P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. Robinson, "Unmanned aircraft system traffic management (utm) concept of operations," in *AIAA Aviation Forum*, 2016.
- ⁷T. Prevot, J. Rios, P. Kopardekar, J. E. Robinson III, M. Johnson, and J. Jung, "Uas traffic management (utm) concept of operations to safely enable low altitude flight operations," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3292.
- ⁸M. Xue, "Uav trajectory modeling using neural networks," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3072.
- ⁹C. Zhang, X. Zhou, H. Zhao, A. Dai, and H. Zhou, "Three-dimensional fuzzy control of mini quadrotor uav trajectory tracking under impact of wind disturbance," in *International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE, 2016, pp. 372–377.
- ¹⁰M. C. P. Santos, C. D. Rosales, M. Sarcinelli-Filho, and R. Carelli, "A novel null-space-based uav trajectory tracking controller with collision avoidance," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2543–2553, 2017.
- ¹¹T. J. Stastny, A. Dash, and R. Siegwart, "Nonlinear mpc for fixed-wing uav trajectory tracking: Implementation and

- flight experiments," in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1512.
- ¹²W. Ren and R. W. Beard, "Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 706–716, 2004.
- ¹³M. Walid, N. Slaheddine, A. Mohamed, and B. Lamjed, "Modeling and control of a quadrotor uav," in *15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. IEEE, 2014, pp. 343–348.
- ¹⁴W. Dong, G.-Y. Gu, X. Zhu, and H. Ding, "Modeling and control of a quadrotor uav with aerodynamic concepts," in *Proceedings of World Academy of Science, Engineering and Technology*, no. 77. World Academy of Science, Engineering and Technology (WASET), 2013, p. 437.
- ¹⁵L. R. Garcia Carrillo, A. E. Dzul Lopez, R. Lozano, and C. Pegard, *Quad Rotorcraft Control - Vision-Based Hovering and Navigation*. Springer-Verlag London, 2013, vol. 1.
- ¹⁶A. Dorobantu, A. Murch, B. Mettler, and G. Balas, "System identification for small, low-cost, fixed-wing unmanned aircraft," *Journal of Aircraft*, vol. 50, no. 4, pp. 1117–1130, 2013.
- ¹⁷R. Beard, "Quadrotor dynamics and control rev 0.1," Brigham Young University, Tech. Rep., 2008.
- ¹⁸A. Vargas, M. Ireland, and D. Anderson, "System identification of multi-rotor uavs using echo state networks," in *AUVSI's Unmanned Systems*, Atlanta, GA, 2015.
- ¹⁹H. Voos, "Nonlinear and neural network-based control of a small four-rotor aerial robot," in *IEEE/ASME international conference on Advanced intelligent mechatronics*. IEEE, 2007, pp. 1–6.
- ²⁰B. S. Kim and A. J. Calise, "Nonlinear flight control using neural networks," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 26–33, 1997.
- ²¹J. Dunfield, M. Tarbouchi, and G. Labonte, "Neural network based control of a four rotor helicopter," in *IEEE International Conference on Industrial Technology*, vol. 3. IEEE, 2004, pp. 1543–1548.
- ²²C. Nicol, C. Macnab, and A. Ramirez-Serrano, "Robust neural network control of a quadrotor helicopter," in *Canadian Conference on Electrical and Computer Engineering*. IEEE, 2008, pp. 001 233–001 238.
- ²³H. Bou-Ammar, H. Voos, and W. Ertel, "Controller design for quadrotor uavs using reinforcement learning," in *IEEE International Conference on Control Applications (CCA)*. IEEE, 2010, pp. 2130–2135.
- ²⁴A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaed, "Flight pid controller design for a uav quadrotor," *Scientific research and essays*, vol. 5, no. 23, pp. 3660–3667, 2010.
- ²⁵J. Escareño, S. Salazar, H. Romero, and R. Lozano, "Trajectory control of a quadrotor subject to 2d wind disturbances," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 51–63, 2013.
- ²⁶S. Waslander and C. Wang, "Wind disturbance estimation and rejection for quadrotor position control," in *AIAA Infotech@ Aerospace Conference*, 2009, p. 1983.
- ²⁷D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.
- ²⁸V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
- ²⁹S.-I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE transactions on Neural Networks*, vol. 3, no. 5, pp. 801–806, 1992.
- ³⁰D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- ³¹S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- ³²G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- ³³B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.
- ³⁴J. Xie, Y. Wan, K. Namuduri, S. Fu, and J. Kim, "A comprehensive modeling framework for airborne mobility," in *AIAA Infotech@ Aerospace Conference*, 2013, p. 5053.
- ³⁵J. Xie, Y. Wan, B. Wang, S. Fu, K. Lu, and J. H. Kim, "A comprehensive 3-dimensional random mobility modeling framework for airborne networks," *IEEE Access*, vol. 6, pp. 22 849–22 862, 2018.
- ³⁶Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.