Deep Learning Methods for System Identification of UAV

Brilian Putra Amiruddin*
Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember

Abstract- Nowadays, deep learning is the most prominent subject in the machine learning field. With the bloom of researchers in this field, numerous novel algorithms are used to solve everyday life problems. The control systems field is one of the subjects that get many impacts of machine learning emergence. System identification of Unmanned Aerial Vehicles (UAV) is one of the control systems problems that could be solved by using deep learning methods. In this paper, Recurrent Neural Networks (RNNs) are applied to identify the system of UAV. Three different models of Deep RNNs have been tried, and the results implied that the RNNs-1 was giving more excellent performance both on the testing MSE and RMSE with the values equal to 0.0006 and 0.0242, successively.

Index Terms- deep learning, systems identification, UAV, quadcopter, recurrent neural networks

I. Introduction

LATELY, both technological advancements and practical applications of Unmanned Aerial Vehicles (UAVs) are growing rapidly. UAV is widely used in real-world applications such as scientific research, civil engineering, military, area mapping, rescue operation, and risk zone inspection [1]. UAV's real-world applications are incredibly vast because every person in this era can easily get their own UAV from the market or create their own UAV as they want. One of the most used UAV types is a quadcopter. From a control systems perspective, the quadcopter is considered the difficult-to-model system due to its nonlinear, unstable, and time-varying characteristics [2]. In line with that, it is arduous to model the quadcopter system using the simple physics law directly. Henceforth, the system identification approach should be used because it provides convenience to get the system's model from the experimentally collected system's data. On the other hand, due to the broadening and emerging machine learning methods implementation, the task of system identification also gets a breath of fresh air approach. The term system identification is usually related to building a statistical model to identify the system's parameters.

Still, since the machine learning methods are receiving substantial attention, this point of view has completely changed. In fact, both the statistical model and the machine learning model itself can be used to identify the system, such as quadcopter. One of the machine learning types that is promising to get the pattern of a system is deep learning. Moreover, nowadays, many deep learning methods are successfully implemented to do pattern recognition and identification of simple systems until the complex one. One example is Deep Recurrent Neural Networks (RNNs) [3]. To the best of the author's knowledge, there are no such works of literature that have compared the performance of different RNNs models for identifying the quadcopter system. Thus, in this paper, some deep learning methods are proposed and utilized to identify the quadcopter UAV system. After the results were obtained, then the models' performance was compared to get the best way of identifying the quadcopter system using deep learning methods.

II. Methods

A. UAV Quadcopter Model

The UAV quadcopter model which used in this experiment is the Parrot Mambo quadcopter, manufactured by Parrot SA. The quadcopter has many different types of sensors including a camera, ultrasonic, and Inertial Measurement Unit (IMU). Basically there are 12 system states relied on the quadcopter system as in (1) [4]

$$X = [x \quad y \quad z \quad \theta \quad \phi \quad \psi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\theta} \quad \dot{\phi} \quad \dot{\psi}] \qquad (1)$$

where $[x \quad y \quad z]$ are the position coordinates of the quadcopter with respect to the global frame, and $[\theta \quad \phi \quad \psi]$ are the Euler angles or orientation of the quadcopter (Pitch, Roll, and Yaw). The dot sign denotes the first differentiation of each system's state.

B. Datasets and System Identification

The experimental dataset of the quadcopter is directly collected using the Parrot Mambo quadcopter model's Simulink simulation environment with the sampling time of 0.005 s by letting the quadcopter follows the orbit trajectory with an altitude of 1 m for more or less 95 seconds. The dataset is not collected directly from the real Mambo quadcopter due to some technical difficulties and issues. The total data gathered from the simulation is equal to 19056 data. The data collected from the simulation was consists of the altitude data, the Euler angles rate data, and thrust data (each thrust and total thrust data).

The system identification task of a UAV quadcopter is divided into two tasks. First, the SISO model identification task which identifies the altitude of the quadcopter by employing total thrust data and the delayed altitude data as the input. Second, the MIMO model identification task that uses delayed Euler angles rate data and four thrusts data as input for identifying the Euler angles rate or velocity (Pitch, Yaw, Roll). The system identification itself is playing a crucial role in control systems development. Without an accurate model, the controller design will be got a problem especially if the developed controller is a model-based controller.

C. Deep Learning Models

There are 3 deep learning models utilized in this experiment, namely RNNs-1, RNNs-2, and RNNs-3. Each model consists of Recurrent Neural Networks (RNNs) with different layers' depth and ended with Fully Connected Layers with Linear Activation Function on the output layer. The RNNs-1 is having 6 stacks of RNNs layers with the hidden units number (500, 300, 200, 100, 50, and 20), RNNs-2 employs 7 stacks of RNNs layers with the hidden units number (650, 400, 300, 250, 100, 50, and 20) and 8 stacks of RNNs with the number of units (750, 500, 400, 350, 200, 100, 50, and 20) on the RNNs-3 model. The input and output layer size depends on the identification task.

For SISO the output layer has 1 output and for MIMO the output layer has 3 output. The RNN models' depth was varied to acquire the effects of the deeper layers towards the identification of UAV systems. The model is trained using Stochastic Gradient Descent optimizer with the learning rate of 10e-4 and the Huber Loss Function for the objective function [5]. Then after the model was trained, the model is validated using 2-Folds Cross-Validation to get each model performance according to the Mean-Squared-Error (MSE) and Root-Mean-Squared-Error (RMSE) metrics. The training batch size number which utilized during training and testing in this experiment is 32.

III. Results and Discussion

The results of the experiment show that the RNNs-1 model achieved the best results both in the training and testing metrics (RMSE and MSE) compared to the other models. The RNNs-1 averaged testing MSE and RMSE are 0.0254 and 0.0007 which is almost zero. This result implies that the RNNs-1 model is excellent in identifying the quadcopter system. Table I shows the averaged RMSE and MSE of all identification tasks. Interestingly, the RNNs-2 and RNNs-3 were also performed very well on every identification task although the performance is not as superb as the RNNs-1 model. Fig.1 shows the results graph of all identification tasks (SISO and MIMO). The plot shows that all of the models were well-fitted to the quadcopter's dataset.

TABLE I
Averaged MSE and RMSE for ALL Identification Cases

Model	Training RMSE	Testing RMSE	Training MSE	Testing MSE
RNN-1	0.0242	0.0254	0.0006	0.0007
RNN-2	0.0300	0.0300	0.0011	0.0011
RNN-3	0.0353	0.0329	0.0015	0.0013

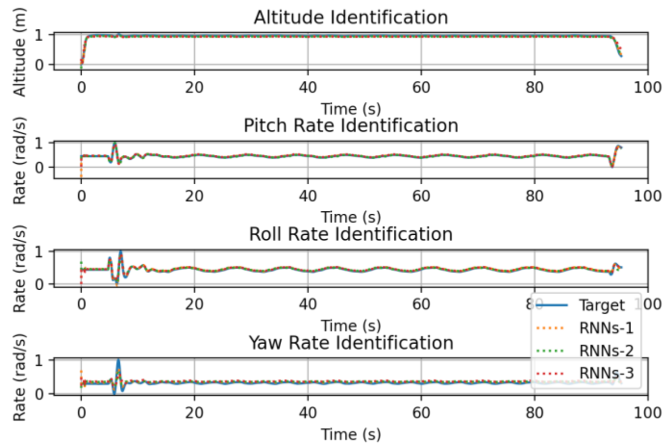

Fig. 1. Quadcopter UAV System Identification Results using the RNNs Models

Furthermore, the results also show that the deeper network did not always perform better than the shallower one. The RNNs-1 only has 6 stacks of RNNs layer and on the other hand, the RNNs-2 and the RNNs-3 are having more stacks of RNNs layers than the RNNs-1 but they did not perform better than the RNNs-1.

IV. Conclusions and Future Work

To sum up, this paper presents the identification of the UAV quadcopter system using several different deep learning methods, precisely the Deep Recurrent Neural Networks (RNNs) with three different models namely, RNNs-1, RNNs-2, and RNNs-3. The experiment shows that the RNNs-1 model is the top-performing model with the minimal both in MSE and RMSE metrics. The results also implied that the deeper networks did not always perform better than the shallower one. Because the performance of the model is very depending on the dataset and the hyperparameters which used not only on the architecture.

In the future, the hyperparameters choosing and tuning for each model will be considered further, so, the learning rate will vary for each model. Also, the identified model from this experiment will be used to design a flight controller for the Parrot Mambo quadcopter using the learning-based control approach and model-based controller, thus the quadcopter can be controlled using our designed controller.

V. References

[1] N. Xuan-Mung and S.-K. Hong, "Improved Altitude Control Algorithm for Quadcopter Unmanned Aerial Vehicles," *Appl. Sci.*, vol. 9, no. 10, p. 2122, May 2019, doi: 10.3390/app9102122.

[2] M. M. Ferdaus, S. G. Anvatti, M. A. Garratt, and M. Pratama, "Fuzzy Clustering based Nonlinear System Identification and Controller Development of Pixhawk based Quadcopter," presented at the International Conference on Advanced Computational Intelligence (ICACI), Doha, Qatar, Jul. 2017, doi: https://doi.org/10.1109/ICACI.2017.7974513.

[3] A. L. Caterini and D. E. Chang, *Deep Neural Networks in a Mathematical Framework*. Cham: Springer International Publishing, 2018.

[4] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Melbourne, Australia, Nov. 2016, pp. 498–503, doi: 10.1109/ICAMechS.2016.7813499.

[5] P. J. Huber, "Robust Estimation of a Location Parameter," p. 27.