A bio-inspired design of real-time control learning algorithm in model-free drone manipulation

Jiaguo Wang[a], Wenheng Li[b], Yang Pei*[a]
[a] Northwestern Polytechnical University, Xi'an, Shaanxi, 710072, China; [b]AVIC Xi'an Aeronautics Computing Technique Research Institute, Xi'an, Shaanxi, 710068, China

ABSTRACT

In recent years, neural-network-based adaptive dynamic models are commonly used to estimate and control flight dynamics for drones and multi-copters. However, most of them simply use networks to optimize few parameters in control policy. There are still large improvements on the structural layout for robust model-free control applications. Therefore, to achieve a similar intelligence of human is still challenging due to their difference in basic mechanisms and difficulty in network modeling. In this paper, we design a control learning algorithm which combines reinforcement learning with neural networks simplified from human cerebellar motor learning model. The algorithm learns parameters by statistically measuring the performance and analyzing the input-output relationship on real-time episodes. In local linear systems, parameters are learned with respect to a spatial function of environment state and subjective expectation. Compared with other methods using static models, the most obvious advantage of this algorithm is that it can learn complex dynamics of alternative degrees of freedom while the dynamics are difficult to be formulated by equation set. Besides, the algorithm is suitable for individual systems, without prior knowledge about system geometry, centre of gravity as well as installation error, since it learns the dynamic effects directly relevant to the optimal guidance and control behavior in unknown or partially known environments instead. Experiment verifies the algorithm in a practical way. In the experiment, the algorithm is implemented to a quad-copter and it can learn the flight control policy from zero-state without any prior knowledge. The flight quality is tested to be equal to accurate control model at outdoor flight experiences. By repeated experiment, the algorithm is demonstrated to have good robustness to control different physical models and the potential to explore alternative dimensionality.

Keywords: Drone, control learning algorithm, reinforcement learning, dynamic models

1. INTRODUCTION

In 21th century, artificial intelligence and the advanced learning technology attracts researchers from widespread areas to improve the horizon of aerospace. As the developing of areas such as drones, aircraft and underwater vehicles, researchers begin to pay more attention to its intelligence, autonomous performance and the adaptation ability to complex environment. Recently, researchers combine mathematical physics methods with models in neuroscience and reinforcement learning to solve problems in autonomous system control.

Motor learning[1-3] is a biological model which describes how neural networks could learn the dynamics of human body. This is the inspiration from the proficient motility of human and animals, especially its nervous system. "Motor" comes from anatomical termination and means motion of muscles[3]. Motor learning can be implemented to robots, drones and any other closed-loop systems interacting with the physical world to perform their tasks with skills learned and memorized. By modeling human CNS (central nervous system), those models help machines take the advantages of human intelligence. Takatori and Inagaki[4] use artificial cerebellar neural network model which they developed to understand motor learning in adaptive motor control. They investigate how gain value was modified context-dependently by neural network. Their mechanism indicates a new method to optimize gain value. Antonietti and Casellato[5] apply artificial cerebellar neural network to a humanoid NAO robot in behavior generation, the robot can respond appropriately at correct timing. Schaal and Ijspeert[6] approach motor learning in a view of statistics in movement imitation tasks. They consider relevant problems such as matching, coordinate transformation, redundancy reduction at single neuron level.

*youngpeinpu@163.com

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), edited by Yang Yue, Proc. of SPIE Vol. 13395, 133951G · © 2024 SPIE · 0277-786X · Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3049899

In recent years, machine learning provides us with a new mathematical aspect to solve problems in clustering, predicting and decision making. In the past, undetermined parameters are selected by empirical formulas or gradient descend. Nowadays, machine learning changes the way of parameter optimization and offers more convenience. Goedhart[7]tries to optimize the gain for PID flight control system of a flapping wing aircraft by k-means cluster classifier, which is famous for many researchers. It is easy to estimate gain configuration where actual dynamic parameters are hard to measure.

In this paper, we propose a useful control learning algorithm from human cerebellar motor learning model combined with reinforcement learning, in which alternative degrees of freedom are represented by linear MIMO reflexes. It arranges partial gain factors in a matrix form. Increase of degrees of freedom only changes matrix dimensions. Further, the learning process is based on the updating of reflex matrix. The algorithm learns appropriate values of each component of reflex matrix by analyzing the statistical relationship between each input and output. The learned parameter matrix is memorized with respect to environment state and subjective movement expectation. Then, Interpolation is used to map the complete control law learned in whole state space so that our model can be implemented to high dimension local-linear systems.

2. EXPERIMENT VALIDATION

2.1 Purpose

To exam the performance of our control learning algorithm, we design an experiment of a flight control task. To test how our learning algorithm can fit different physical models of other degrees of freedom, we applied the algorithm to a quadcopter in an experiment. After hanged on a string, the quadcopter can control its balance in the air. Both results have visualized eigenmovement matrix $\frac{k}{\|k\|}$, which is easy to understand and helpful to verify.

2.2 Experiment: balance of a quadcopter

The quadcopter (in Figure 1) (about 18cm * 18cm * 8cm) contains 4 motors and an IMU. Before the experiment, the quadcopter does not have the ability to fly in the air. After the quadcopter is hanged on a string while learning the control parameters of eigenmovement matrix $\frac{k}{\|k\|}$, it can obtain this control ability. In the quadcopter, the 4 motors randomly link to the 4 connectors on its central circuit board. But the algorithm does not require the sequence, electrical connection and system geometry. The algorithm can demonstrate a non-model based learning. On the other hand, compared to experiment, the algorithm can also demonstrate a free dimensionality.

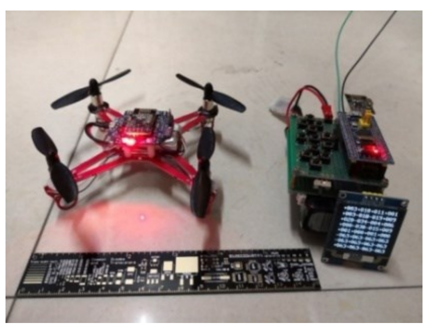


Figure 1. The quadcopter and remote controller.

Figure 2 is the control flowchart of the quadcopter. Except for the learning module, the whole control structure forms an adaptive feedback control. We sample the gyroscope data and accelerometer data at 100Hz from IMU to get rotational velocity p, q, r and gravity vector ax, ay, az. Madgwick[8] is implemented to derive Euler angles, which comes to the secondary sensory input vector I_2. Then, I_2 compares to the remote controller signals from player. The controller signals E_2 indicate the expected Euler angles of the quadcopter. The difference between I_2 and E_2 comes to the expected rotational velocity ep, eq, er. Again, the difference of ep, eq, er between p, q, r comes to the input of reflex module and learning module. The learning module monitors the input and output of reflex module and updates on the eigenmovement matrix $\frac{k}{\|k\|}$ of reflex module. The output of reflex module drives 4 motors respectively.

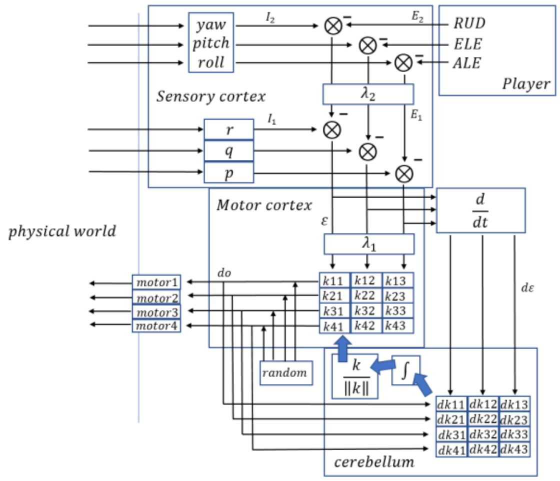


Figure 2. Schematic diagram of the quadcopter. (RUD) rudder. (ELE) elevator. (ALE) aileron. (λ_1, λ_2) gains.

2.3 Experiment: method

Before experiment, we check the electrical connection between the 4 motors and the central circuit board, and we infer the possible learning result of eigenmovement matrix $\frac{k}{\|k\|}$ in Table 1, which wholly depends on the position of motors in perfect condition (no installation error, no damage). The column "p" "q" "r" indicates three perpendicular directions in Euler angles.

Table 1. Expected convergent value of $\frac{\boldsymbol{k}}{\|\boldsymbol{k}\|}$.

Motor index	r	q	p
1	-0.5	-0.5	0.5
2	0.5	0.5	0.5
3	-0.5	0.5	-0.5
4	0.5	-0.5	-0.5

Experiment contains training mode and testing mode. In the training mode, the quadcopter is hanged on a string (Figure 3a) to simulate the environment in the air, since it does not have the skill to balance itself in the air at the beginning. Reflex module is added with a random generator to explore the space. Covariance matrix k is initialized with unit vectors before the training starts. During training, random output is sent to motors and propellers. The quadcopter begins to move randomly. Change of rotational velocity is sensed by the IMU, while the learning process keeps running, $\frac{k}{\|k\|}$ keeps updating as time runs. Although being hanged by a string is the easiest way to simulate the condition in the air, it is hard to get rid of the small moment of the string. To reduce the influence of the moment, we try to avoid large oscillation by reducing the overall amplitude of the random output. Within few minutes, we notice that the eigenmovement matrix $\frac{k}{\|k\|}$ come to a convergence (Figure 4). The learned parameters indicates that the quadcopter can obtain a certain level of flight control, but not the best. In the testing mode (Figure 3b), we make the quad-copter fly in the sky to fine-tune the eigenmovement matrix. When controller command is sent to the quadcopter, the quadcopter can quickly move around as the player command. We record the flight with Euler angles for further analysis.

(a)Learning mode: hanged by a string (b)Testing mode: flying and fine-tuning

Figure 3. Experiment environment.

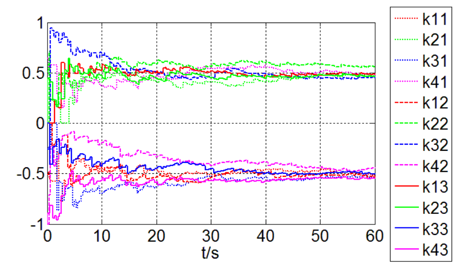


Figure 4. Convergence of $\frac{k}{||k||}$.

3. RESULTS AND CONCLUSION

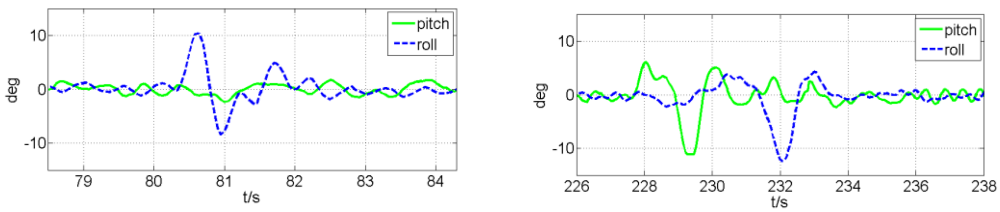
The quadcopter successfully flies in the air as we expected, without previous knowledge about the geometry of the structure, including position of motors, sequence of motors connected. Except for some offsets, the averaged convergence value (Table 2) is basically the same as our expectation in Table 1. Issues such as installation error, centre of gravity, dust and hair twined on the axle of motors and damage of propellers would generate an offset of the convergent value. The reason why the convergent value does not come to Table 1 is that the influence of such issues is included in Table 2. Therefore, the convergence value in Table 2 should be roughly around the value in Table 1. Table 1 can be a reference but may not be the accurate value of individual aircraft. This control learning algorithm can approach to the precise value for individual theoretically, but it is difficult to measure and hard to evaluate in the experiment.

Table 2. Convergent value of $\frac{\boldsymbol{k}}{||\boldsymbol{k}||}$.

Motor index	**r**	**q**	**p**
1	-0.4514	-0.4523	0.4433
2	0.4910	0.4523	0.5224
3	-0.5386	0.5475	-0.5383
4	0.5148	-0.5396	-0.4908

Manipulation from player can be carried out efficiently. To find out how these offsets would affect our flight quality, we estimate the stability of the quadcopter. The command of remote controller plays a role in changing the expected Euler angles, hence, a pulse in joystick can be viewed as a kind of disturbances. Here we compare the stability between using eigenmovement matrix in Table 1 and Table 2. First, we use the matrix in Table 2. When the quadcopter is flying in the air, a pulse on aileron joystick is recorded at roughly 80.3s, the Euler angles are shown in figure 5a. The oscillation of roll angle reaches 10 degrees and come back to previous condition (1.5 degrees) within 2 seconds. Next, we use the matrix in Table 1. When the quadcopter is flying in the air, a pulse on elevator joystick is recorded at roughly 228s, followed by a pulse on aileronat roughly 231s, the Euler angles are shown in figure 5b. The oscillation of roll angle reaches 12 degrees and come back to previous condition (1.5 degrees) within 2 seconds. The configuration of λ_1 and λ_2 are the same, both 2 trials have 1.5 degrees oscillation before we move joystick. No obvious difference in stability is found. Both 2 trials take

2 second to recover, they both have the same level of overshoot. The pattern is similar during all fliying test, hence, the convergent value in Table 2 is applicable for this quadcopter.



(a) Using $\frac{k}{\|k\|}$ in Table 2. (b) Using $\frac{k}{\|k\|}$ in Table 1

Figure 5. Stability test.

This paper proposes a control learning algorithm simplified from human cerebellum. It learns the skills by statistically analyzing the input-output relationship. The most obvious advantage for this control learning is that it can learn the skills between complex input and output. To derive dynamics equations in other methods would be difficult while our control learning algorithm can work more efficiently. Another strength for this algorithm is it can find suitable parameters for individual systems. Issues such as installation errors are hardly to avoid in the previous work. By contrast, this algorithm requires no geometry and electrical connections of the system as well as the installation errors since it directly learns the control parameters instead. Parameters can fit each different individual. Also, the algorithm is organized in option and condition space, which gives the ability to adapt to the changing of environment.

An experiment is designed to demonstrate the algorithm in a practical way, and our control learning algorithm shows good performance in different models. In the experiment, the algorithm helps a quadcopter learn to control flight in the air. The experiment results fit individual models in detail. We change the degrees of freedom and demonstrate that the algorithm only assumes the local linear dynamics and has no limitation on dimension. Therefore, it can learn over alternative degrees of freedom. In complex tasks, the algorithm has the potential to explore high dimensionality.

ACKNOWLEDGMENTS

The present work is supported by the National Natural Science Foundation of China (Project No.12172287). I would like to show my gratitude to my colleague, Dr. Zhong Yun Fan for providing technical assistance.

REFERENCES

[1] Wolpert, D. M., "Computational approaches to motor control," Trends in Cognitive Sciences, 1(6), 209-216 (1997).
[2] Jordan, M. I., "Computational aspects of motor control and motor learning," Handbook of perception and action, 71-120 (1996).
[3] Thomas, J., Siegelbaum, S. and Hudspeth, A. J., [Principles of Neural Science], New York: McGraw-Hill (2000).
[4] Takatori, S., Inagaki, K. and Hirata, Y., "Realization of direction selective motor learning in the artificial cerebellum: simulation on the vestibuloocular reflex adaptation," 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 5077-5080 (2018).
[5] Antonietti, A., Casellato, C., D'Angleo. E., et al, "Bioinspired adaptive spiking neural network to control NAO robot in a Pavlovian conditioning task," 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), 142-147 (2018).
[6] Schaal, S., Ijspeert, A. and Billard, A., "Computational approaches to motor learning by imitation," Philosophical Transactions of the Royal Society of London B: Biological Sciences, 358(1431), 537-547 (2003).
[7] Goedhart, M., Van Kampen, E. J., Armanini, S. F., et al, "Machine learning for flapping wing flight control," 2018 AIAA Information Systems-AIAA Infotech@ Aerospace, (2018).
[8] Madgwick, S. O., Harrison, A. J. and Vaidyanathan, R., "Estimation of IMU and MARG orientation using a gradient descent algorithm," Rehabilitation Robotics (ICORR), 1-7 (2011).