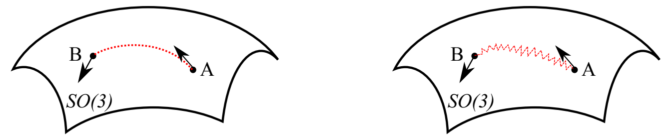
2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

High Performance Full Attitude Control of a Quadrotor on SO(3)

Yun Yu, Shuo Yang, Mingxi Wang[1], Cheng Li, Zexiang Li[2]

Abstract— **This paper presents a novel quadrotor UAV attitude control algorithm to realize complex acrobatic UAV maneuvers. A nonlinear dynamic model based on the exponential coordinates parametrization of rotation is proposed. By analysing the model using Lie Group and Lie Algebra theory, cascaded linear PID controllers are designed. To further improve the controller performance, PID controllers are augmented with smith predictor and rotational trajectory planner. The experiments conducted on a real quadrotor show that our control algorithm surpasses most known quadrotor controllers.**



(a) Smooth trajectory (b) Trajectory of separate control

Fig. 1: Comparsion of trajectory of rotation using two different control methods.

I. Introduction

In past 10 years, small scale Unmanned Aerial Vehicles (UAV), represented by multirotor UAVs, have received considerable attention from robotic and aeronautic research community. By 2005, researchers have already devised several stable UAV control algorithms [3], [6], [9]. After 2010, people focus more on multi robot formation research [15], [16] or vision based UAV navigation [4], [18]. By the end of year 2012, more than 10 quadrotor products are commercially available, including Parrot ardrone, AscTec Pelican, and DJI Phantom. Since its wide commercialization, all aspects of multirotor UAV are extensively studied.

However, recent heavy application level demands on multirotor UAV let authors rethink the multirotor UAV control problem, and it was found that several issues are not properly addressed by conventional commercial multirotor controllers.

In conventional methods, attitude control is not treated as a geometric tracking problem. Most existing methods use Euler Angles to parametrize the rotation of a UAV, and yaw-roll-pitch angles are separately controlled with separated target angles [7], which linearize the rotation of UAV. This approach inherently neglects the manifold structure of rotation. When the UAV is rotating, it is actually moving from one point to another on SO(3). All movements on SO(3) must take the manifold structure into account as velocity constraints. A smooth movement trajectory is prefered. However, separate control of yaw-roll-pitch angles results in a non-smooth trajectory, because it is only the local approximation of SO(3). As illustrated in Figure 1. Moreover, if point A and B are far from each other, separate control may not be able to decide target yaw-pitch-roll angles because of gimbal lock problem of euler angles.

As a consequence, separate control limits the maneuverability of UAVs. Complex UAV maneuver controls such as throw start and flipping in the air involve consequent target attitude tracking and path planning on SO(3). The movements must be fast and along smooth trajectories. Without a proper treatment of SO(3), these types of control cannot be realized steadily.

Separate control is widely accepted for two reasons. One is that most multirotor controllers can only regulate UAV orientation to hover orientation, and their angular deviations are usually small. In the vicinity of the hover orientation, separate control is a precise enough approximation. Another reason is most of existing multirotor UAVs are small quadrotors. Main stream multirotor UAV researches focus on UAVs weight less than 1kg [11]. The problem with separate control is not obvious on such small UAVs, as their agilities cover control defects. But for large UAVs, such as DJI S800 (weight 5kg-8kg) and S1000(weight 7kg-12kg) [2], separate control cannot guarantee good performance when doing large angle movements and complex maneuver behaviors. To our best knowledge, no existing multirotors heavier than 1kg support flipping in the air and throw-to-start function.

In this paper we view attitude control as a SO(3) tracking problem. The rotation of UAV is rigorously analyzed to derive the control law. Additionally we also considered rotor dynamics from an engineering approach. Actuator performance is known to be one critical design factor of multirotor UAVs. We proposed a practical method to model rotor dynamics, eliminating the need of a precise rotor model. To compensate the slow response time of rotors, we add smith predictors to the PID control loops.

This paper is organized as follows. We first provides necessary mathematical tools that are used to derive control law in Section III. Then we discuss the control law in SectionIV. SectionV present implementation details, experiment results and demo video link. SectionII does a brief literature review. Finally SectionVI concludes the paper.

II. Related Works

In literature, PID control on SO(3) is well studied. Researchers in robot manipulation and helicopter control al-

[1]Yun Yu, Shuo Yang, Mingxi Wang are with DJI Innovations and the Department of Electronic & Computer Engineering, Hong Kong University of Science & Technology {yyuao, syangag, mwang}@ust.hk
[2]Cheng Li, Zexiang Li are with the Department of Electronic & Computer Engineering, Hong Kong University of Science & Technology {cliac, eezxli}@ust.hk

978-1-4799-6923-4/15/$31.00 ©2015 IEEE

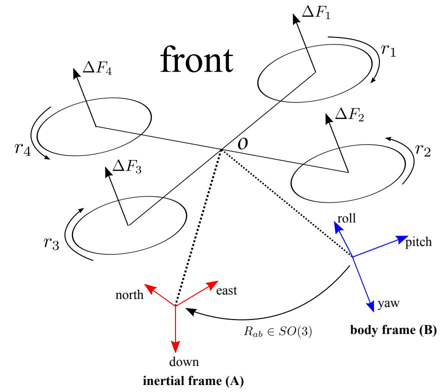


Fig. 2: The model of the aerial robot. Red frame is the inertial frame and blue frame is the body frame. Both frames ought to coincide with the origin O. They are separated here only to illustrate axis directions.

ready agreed that "error measure should correspond to the topology of the error space" [19], which lead us to examine the disadvantage of separately pitch-roll-yaw control. Our derivate of UAV control law are mainly inspired by [19] and [8]. The proof of theorems and propositions in our work can be found in [8].

In UAV research community, the only work that considers the property of SO(3) is [12]. A different approach is used to derive the control law, which is similar to ours. However, they did not implement the controller on a real UAV. Only simulation results are presented. Their work is adopted in [13] for trajectory control of small UAVs. Compared with our work, the focus of [13] is micro UAV control. Their work deals with UAVs that are small and light weight.

III. Mathematics Preliminaries

In this section we define our notation for rotation parametrization and review some basic knowledge of Lie Group and Lie Algebra.

A. Rotation Representation

As shown in Figure 2, if we denote the inertial frame as A and B the body frame. We can represent the bases of frame B using the axes of frame A, with coordinates $\mathbf{x}_{ab}$, $\mathbf{y}_{ab}$, $\mathbf{z}_{ab} \in \mathbb{R}^3$. Then we define matrix

$$R_{ab} = [\mathbf{x}_{ab} \quad \mathbf{y}_{ab} \quad \mathbf{z}_{ab}]$$

as the rotation matrix from frame B to frame A. For any points q_b in frame B, its coordinates in frame A is

$$q_a = R_{ab} q_b$$

B. Rotation Decomposition

All possible rotations form the rotation group of $\mathbb{R}^3$, namely $SO(3)$. The group structure of SO(3) leads to composition rule for rotations. If we have three frames A, B and C. Then we have

$$R_{ac} = R_{ab} R_{bc}$$

Therefore, given frame A and frame C, it is always possible to choose an intermediate frame B such that R_{ac} can be decomposed into R_{ab} and R_{bc}.

C. Exponential Map & Logarithmic Map

According to Euler's Rotation Theorem, any element R of SO(3) is equivalent to an axis-angle rotation with axis $\omega \in \mathbb{R}^3$ and angle $\theta \in [0, 2\pi)$. The canonical representation of $R \in SO(3)$ is exponential coordinates [17]

$$R = e^{\hat{\omega}\theta} = I + \theta\hat{\omega} + \frac{\theta^2}{2!}\hat{\omega}^2 + \frac{\theta^3}{3!}\hat{\omega}^3 + \dots$$

where $\omega = [\omega_1 \quad \omega_1 \quad \omega_3]^T \in \mathbb{R}^3$ with $\|\omega\| = 1$ and the $^\wedge$ operator is defined as

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

And the inverse operation of the $^\wedge$ operator, denoted as $^\vee$, recovers ω from $\hat{\omega}$.

$\hat{\omega}\theta$ is skew-symmetric matrix satisfies $\hat{\omega}^T\theta = -\hat{\omega}\theta$. All such matrices forms a vector space denoted as $\mathfrak{so}(3)$:

$$\mathfrak{so}(3) = \{S \in \mathbb{R}^{3\times3} : S^T = -S\}$$

So we have the exponential map $exp : \mathfrak{so}(3) \to SO(3)$ from $\hat{\omega}\theta$ to $R = e^{\hat{\omega}\theta}$. We can prove that

$$e^{\hat{\omega}\theta} = I + \hat{\omega} sin(\theta) + \hat{\omega}^2(1 - cos\theta)$$

The inverse map from SO(3) to $\mathfrak{so}(3)$ is the logarithmic map, which is defined as

$$log(R) = (R - I) - \frac{(R - I)^2}{2} + \frac{(R - I)^3}{3} - \dots$$

From the properties of the exponential map, it can be constructively proved that

$$log(R) = \frac{\phi}{2sin\phi}(R - R^T) \qquad \in \mathfrak{so}(3)$$

in which ϕ is $cos^{-1}(\frac{tr(R)-1}{2})$ and $|\phi| < \pi$. If $R = I$, ω can be chosen arbitrarily.

D. Properties of SO(3)

According to Lie Group Theory, SO(3) is a typical compact Lie group with corresponding Lie algebra $\mathfrak{so}(3)$. For any $g \in SO(3)$ and $X, Y \in \mathfrak{so}(3)$, we define the adjoint map Ad_g and lie bracket ad_X to be

$$Ad_g(Y) = gYg^{-1}$$
$$ad_X(Y) = [X, Y] = XY - YX$$

On SO(3), the Killing form $\langle\cdot,\cdot\rangle_K$, which is a bilinear operator on $\mathfrak{so}(3) \times \mathfrak{so}(3)$, is defined as

$$\langle X, Y\rangle_K \triangleq tr(XY)$$

Then the inner product of $\mathfrak{so}(3)$ defined base on the Killing form is $\langle\cdot,\cdot\rangle \triangleq -1/4\langle\cdot,\cdot\rangle_K$. This inner product is *Ad-invariant*:

$$\langle X, Y\rangle = \langle Ad_g X, Ad_g Y\rangle, \qquad \forall g \in SO(3)$$
$$\langle ad_Z X, Y\rangle = -\langle X, ad_Z Y\rangle, \qquad \forall Z \in \mathfrak{so}(3)$$

Inner product defines the metric property of SO(3), then we have one proposition and one theorem [5].

Proposition 1. *For Lie group G, the distance between an element g and the identity $I \in G$ is given by the norm of the logarithmic map:*

$$\|g\|_G = \langle log(g), log(g)\rangle^{1/2}$$

Consider a dynamic system on SO(3) has state $g \in SO(3)$. Then this system has state transition function as

$$\dot{g} = gV^b = V^s g, \quad V^b, V^s \in \mathfrak{so}(3) \tag{1}$$

where V^b and V^s stands for velocity viewed in body frame and inertial frame respectively. A detailed proof to this fact is in [17].

Theorem III.1. *(Derivative of distance function) Let G be a compact Lie Group with inner project $\langle \cdot, \cdot \rangle$. Consider a smooth trajectory $g(t) \in G$ that never passes through a singularity of the exponential map. Then*

$$\frac{1}{2}\frac{d}{dt}\|g\|_G^2 = \langle log(g), V^b\rangle = \langle log(g), V^s\rangle$$

We give all definitions and propositions without proof. Readers may refer to [5], [10], [17] for detailed treatment to Lie Group and Lie Algebra.

IV. Attitude Control on SO(3)

We assume that attitude controller has continuous high frequency measurements of the orientation and angular velocities of the UAV. Additionally, angular speed velocities must be filtered in order to generate noise-less angular acceleration measurements. Usually these measurements can be obtained by certain IMU algorithm. The algorithm that generates such measurements is outside the scope of this paper.

A. Model of Aerial Robot Kinematics & Dynamics

Our robot model is a standard quadrotor system contains four identical rotors and two sets of propellers. Each propeller generates torque ΔF_i perpendicular to the plane on which the robot base resides. We choose the local north-east-down (NED) coordinate frame as the inertial frame, and the roll-pitch-yaw coordinate frame as the body frame. Since we do not consider translation in this case, both the inertial frame and the body frame have their origins located at the center of mass of the robot. Our aerial robot has X shape so roll axis equally divide the angle between the arm of motor 1 and the arm of motor 4. Directions of frame axes are shown in Figure 2.

Based on this model, we define:

$R \in SO(3)$, the rotation of the aerial robot expressed as frame transform from the body frame to the inertial frame;

$J \in \mathbb{R}^{3\times 3}$, the inertia matrix expressed in the body frame;

$\omega^b \in \mathbb{R}^3$, the angular velocity in the body frame;

$\tau \in \mathbb{R}^3$, the control torque generated by the actuators of the robot, expressed in body frame as well.

With above definition, we present robot kinematic as a first order system

$$\dot{R} = R\hat{\omega}^b \tag{2}$$

which is rewritten from (1).

Then by Euler equation we can write the robot dynamics as

$$J\dot{\omega}^b + \omega^b \times J\omega^b = \tau \tag{3}$$

B. Control of The Aerial Robot

The control goal of our algorithm is to move from the current rotation R_c to a target rotation R_t. This is a tracking problem. If we regard target rotation as identity $I \in SO(3)$, which coincides with the hover stable position, and let $R_e = R_t R_c^{-1}$ be the rotation from identity position to current position, then the tracking problem is converted to a regulating problem.

Given (2) and (3), we have a second order system of multirotor UAV as

$$\begin{cases} \dot{R} = R\hat{\omega}^b \\ J\dot{\omega}^b = -\omega^b \times J\omega^b + \tau \end{cases} \tag{4}$$

The control input is total torque τ applied to the UAV. To derive the control input we state following theorem without proof [8].

Theorem IV.1. *Consider system on SO(3)*

$$\begin{cases} \dot{g} = g\hat{V}^b \\ \dot{V}^b = f(g, V^b) + U \end{cases} \tag{5}$$

and let K_p and K_d be symmetric, positive definite gains. Then the control law

$$U = -f(g, V^b) - K_p log(g) - K_d V^b$$

exponentially stabilizes the state $g \in SO(3)$ from any initial condition $tr(g(0)) \neq -1$ and for all K_p and $V^b(0)$ such that

$$\lambda_{min}(K_p) > \frac{\|V^b(0)\|^2}{\pi^2 - \|g(0)\|_{SO(3)}^2}$$

where $\lambda_{min}(K_p)$ is the minimum eigenvalue of K_p

The stability of the control law can be confirmed by Lyapunov's second method for stability. For our system in (4), if the Lyapunov function is chosen as

$$W = \frac{k_p}{2}\|g\|_{SO(3)}^2 + \frac{1}{2}\langle \omega^b, J\omega^b\rangle_{\mathbb{R}^3}$$

then the stable control law is

$$\tau = \omega^b \times J\omega^b - K_p(log(R))^\vee - K_d\omega^b \tag{6}$$

where $^\vee$ is the inverse operation of the $^\wedge$ operator.

We further observe that $log(R)$ can be viewed as a target rotation angle $\hat{w}_d^b \in \mathfrak{so}(3)$. Since $\mathfrak{so}(3)$ is a vector space, the second term and the third term in (6) can be separately controlled. We do a sight modification to the control law to get following complete control scheme:

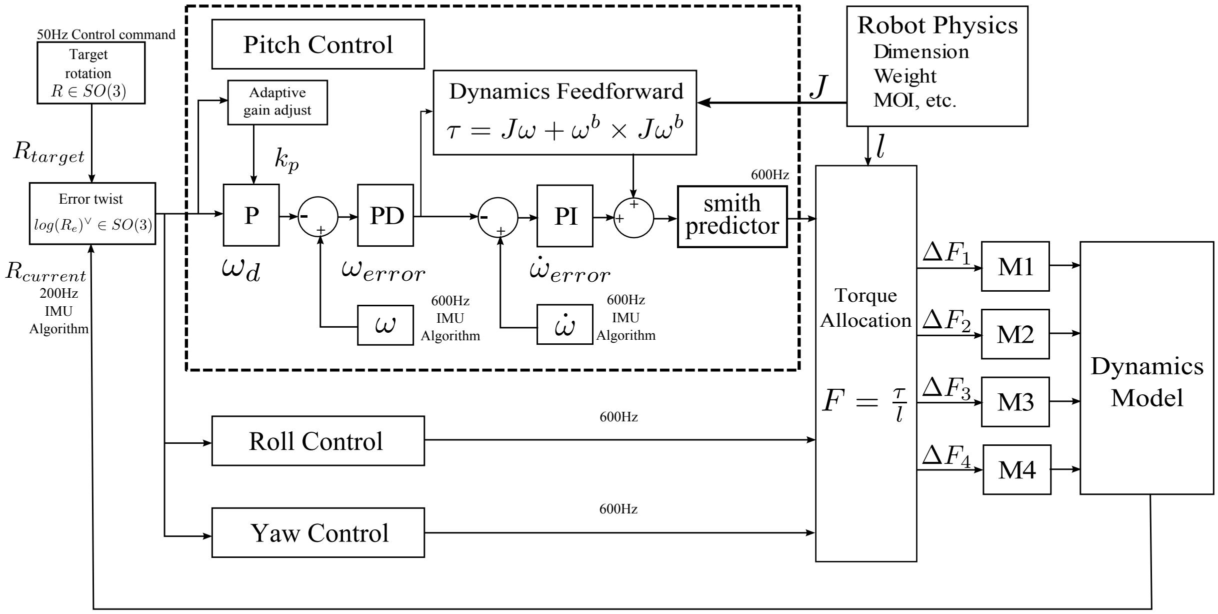


Fig. 3: The complete attitude control diagram. Pitch, roll and yaw control algorithms are the same, so the figure only describes pitch control detail.

Angular velocity control

$$\omega_e^b = (log(R_e)^{\vee} - \omega^b)$$

$$\dot{\omega}_d^b = K_p \omega_e^b + K_d \frac{d}{dt} \omega_e^b$$

Angular acceleration control

$$\dot{\omega}_e^b = (\dot{\omega}_d^b - \dot{\omega}^b)$$

$$\tau = K_p \dot{\omega}_e^b + K_i \int_0^t \dot{\omega}_e^b dt + \omega^b \times J\omega^b$$

all PID gains are adjusted to meet the condition of Theorem IV.1. Although on SO(3) there is one singularity point where $tr(R) = -1$ (the UAV is upside down at this point), the control law will not degenerate. Our experiment shows that even when the UAV reaches the singularity point during flight, it can still recover to the hover position.

Figure 3 summarizes the entire control scheme. Details of engineering units such as adaptive gain adjust, craft physical and dynamical model are omitted. This controller is fundamentally different from conventional controllers, although we still call pitch, roll and yaw control. The most crucial point is that the target angular velocities are obtained in a coupled manner with SO(3) manifold constraints satisfied. So the movement of UAV orientations forms a smooth curve on SO(3). The angular velocity can be separately controlled by three PID controllers because $\mathfrak{so}(3)$ is a vector space, so the movement on this space can have independent coordinate changes. Conventional multirotor controllers exploit the second property but miss the important first one.

C. Attitude Path Planning on SO(3)

Attitude control can be viewed as a path planning problem on SO(3). Planning is necessary because multirotor UAVs is not intrinsically homogeneous. The torque on pitch and roll axes are controlled by speed difference of opposite blades. The rotors only need to change speed a little bit to get enough torque. But for yaw control, the rotors must change significantly to generate needed rotational torque. Some pitch and roll control is always applied faster than yaw control. In our experiments we found that the response time of pitch or roll controller to 10° is around 60ms, but the yaw controller will take more than 150ms to rotate 10° around the yaw axis. Therefore, we add a planner to process target rotation. For any target rotation input either by user command or controller regulator, we calculate the error rotation $R_e = R_t R_c^{-1}$, and decompose R_e as:

$$R_e = R_{torsion} R_{tilt}$$

as plotted in Figure 4.

The calculation of R_{tilt} takes the inner product and cross product of the yaw axes of the current frame and the target

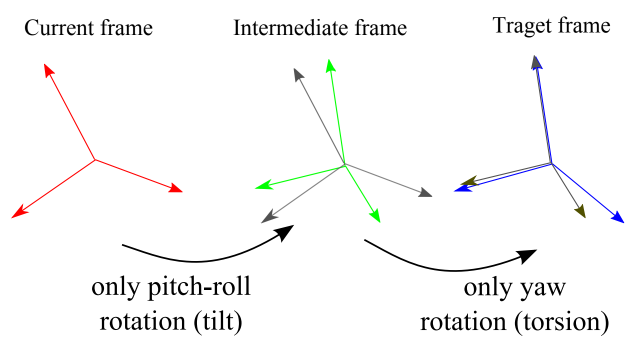
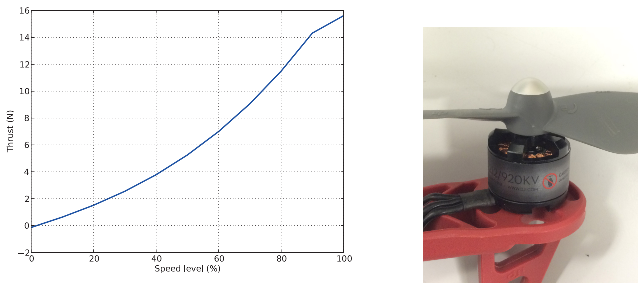


Fig. 4: Tilt-torsion decomposition. For any current (Red) and target (Blue) rotations, add an intermediate (Green) rotation. Then execute a two-stage control action.



(a) (b)

Fig. 5: Relationship between the propeller speed and the thrust force is neither linear nor quadratic. The tested motor is DJI E300.

frame. Let $\mathbf{z}_c = \mathbf{z}_t = [0\ 0\ 0]^T$ be the yaw axis of the target frame, then $R_e^T \mathbf{z}_t$ is the coordinate of this yaw axis in the current frame. The cross product of $\mathbf{z}_c$ and $R_e^T \mathbf{z}_t$ is the rotation axis $\mathbf{r}$ of R_{tilt}. And the angle ϕ between this two axes is the rotation angle. Hence,

$$R_{tilt} = e^{\hat{\mathbf{r}}\phi}$$

The purpose of R_{tilt} is to align the yaw axis of the current frame with yaw axis of the target frame. When UAV controller first take R_{tilt} as input, only pitch and roll controller is doing tracking controller while yaw controller is doing regulating control. After R_{tilt} is reached, tracking of $R_{torsion}$ involves yaw controller. This decomposition decouples the controller behaviors to sequentially execute fast response motion and slow response motion.

With this decomposition, UAV rotation is moving on a planned trajectory on SO(3). From current position, the UAV will first move to the intermediate position, then rotate to the target position.

D. Rotor Dynamics

The last step of the controller is distribute the torques to rotors. Torques have to be converted to propeller rotational speeds and further to ESC signals. To avoid over simplification of the rotor dynamics while have a neat conversion process, we conducted experiments to measure a speed-force curve for one type of DJI motors, as shown in Figure 5a. The curve is stored as a look-up table in the algorithm implementation, with unknown points interpolated from neighboring existing points. For a desired thrust force, we check the corresponding speed level from the table, and apply ESC signal to let rotor change to that rotational speed.

Slow rotor dynamics results in time delay of rotor response, which is critical when the UAV is rapidly adjusting its position. The delay of pitch controller response to step target is around 60ms. This time delay is modeled as a first order transfer function:

$$G_p(s) = \frac{e^{-\tau s}}{T_\phi s + 1}$$

To reduce the time delay, we add a smith predictor to the final step of PID controller. Smith predictor is an effective

Fig. 6: The experimental platform

augmentation to standard PID controller [14]. The smith predictor reduced the delay time by about 15% in our system. However, because of control output saturation, the time delay cannot be further reduce. Ideal rotor has no speed limitation, thus the system delay can be compensate by forward the control output to high value. However, real rotor does has maximum speed, so the system delay is inevitable.

V. Experiment Results

Our experiment platform is a self-made UAV with carbon fiber frame. Its main specifications are listed below.

Takeoff Weight	1150g - 1450g
Rotor Shaft Distance	450mm
Blade size	9 inches

On this platform we measured some important parameters such as rotor delay. The pitch and roll rotor delay is 60ms, and the yaw rotor delay is 150ms. Also we measured the relationship between the propeller speed and the thrust force.

We implemented our control algorithm on the hardware platform of opensource flight controller APM [1]. The original attitude control algorithm in APM, which is a typical separate control algorithm, is used as the baseline method.

We simulate our controller and original APM controller in MATLAB. Their control performance can be compared by step response. A step response is generated to simulate a control command that make pitch angle reach 40 degree. In Figure 7, blue line is the step function, brown line is the response of APM controller. Green line is a theoretical limit if the system delay is known precisely. Purple line is the performance of ideal rotor with no speed saturation limit. Orange line is our controller. Our controller clearly outperforms APM controller in terms of response speed and overshoot curbing.

We also carried real flight experiment in outdoor environment. Figure 8 displays the control error during hovering. The figure shows that the maxmium pitch error and yaw error are both below 1 degree. Therefore the attitude control effect of the UAV during hovering is nearly unobservable.

In these experiments we let the UAV flip in the air and perform throw-to-start function. The video of the UAV performance can be viewed at https://www.youtube.com/watch?v=k6MX3KqZq-A.

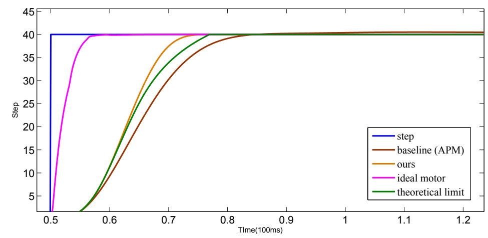


Fig. 7: Step response of our controller

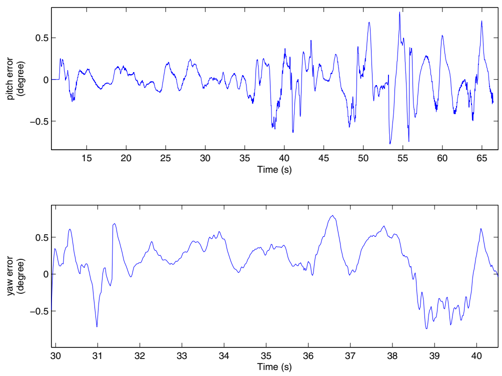


Fig. 8: Angle errors of our controller

VI. Future Work & Conclusion

In this paper we presented a novel quadrotor control algorithm with rotation modeled using exponential coordinates. Control target is planned by considering the property of second order dynamic system on SO(3). Thus the PID controller is the direct controller on $\mathfrak{so}(3)$. Our control algorithm out performs conventional algorithms in terms of response time and complex maneuver behaviors.

The presented attitude control algorithm will be modified and used in future DJI products. As quadrotor attitude control is changing from research topic to standard technology, the future modification of the algorithm will be product-oriented.

Acknowledgment

This research is supported by DJI Innovations and Automation Technology Center of HKUST. DJI engineers contributed a lot of work to create the hardware platform for our control algorithm. Authors would also like to thank Dr. Yuanqing Wu for providing helpful suggestions and valuable insights.

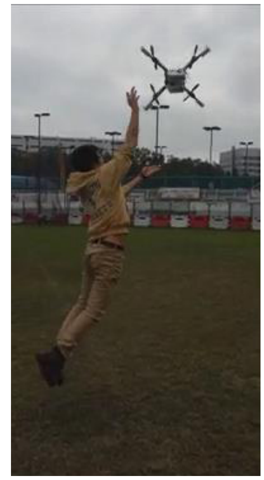


Fig. 9: Our UAV can be thrown up and start to hover autonomously

References

[1] APM Autopilot Suite `http://ardupilot.com/`

[2] DJI S1000 Professional Octorotor `http://www.dji.com/product/spreading-wings-s1000-plus`

[3] Erdinc Altug, James P Ostrowski, and Robert Mahony. Control of a quadrotor helicopter using visual feedback. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 72–77. IEEE, 2002.

[4] Michael Blosch, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Vision based mav navigation in unknown and unstructured environments. In *Robotics and automation (ICRA), 2010 IEEE international conference on*, pages 21–28. IEEE, 2010.

[5] William M Boothby. *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Academic press, 1986.

[6] Samir Bouabdallah, Pierpaolo Murrieri, and Roland Siegwart. Design and control of an indoor micro quadrotor. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4393–4398. IEEE, 2004.

[7] Samir Bouabdallah and Roland Siegwart. Full control of a quadrotor. In *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, pages 153–158. IEEE, 2007.

[8] Francesco Bullo. Proportional derivative (pd) control on the euclidean group. In *European Control Conference*, volume 2, pages 1091–1097.

[9] Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, 25(6):45–55, 2005.

[10] Brian Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2003.

[11] Vijay Kumar and Nathan Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 2012.

[12] Taeyoung Lee, M Leoky, and N Harris McClamroch. Geometric tracking control of a quadrotor uav on se (3). In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5420–5425. IEEE, 2010.

[13] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, page 0278364911434236, 2012.

[14] C Meyer, DE Seborg, and RK Wood. A comparison of the smith predictor and conventional feedback control. *Chemical Engineering Science*, 31(9):775–778, 1976.

[15] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The grasp multiple micro-uav testbed. *Robotics & Automation Magazine, IEEE*, 17(3):56–65, 2010.

[16] Mark Muller, Sergei Lupashin, and Raffaello D'Andrea. Quadrocopter ball juggling. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 5113–5120. IEEE, 2011.

[17] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[18] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 20–25. IEEE, 2011.

[19] JT-Y Wen and Kenneth Kreutz-Delgado. The attitude control problem. *Automatic Control, IEEE Transactions on*, 36(10):1148–1162, 1991.