2016 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 7-10, 2016. Arlington, VA USA

Fractional Order Robust Visual Servoing Control of A Quadrotor UAV with Larger Sampling Period

Bo Shang*, Jianxin Liu, Tiebiao Zhao, YangQuan Chen

Abstract— **Unmanned aerial vehicles (UAVs) are widely applied in both civil and military fields, such as rescue, surveillance, exploration, navigation, precision agriculture and etc., because of small size, low cost and easy maintenance. However the autonomous flight of UAVs under unstructured environment is still open, especially when GPS is unavailable or indoor task is scheduled. Furthermore, GPS-based navigation accuracy needs to be improved. To deal with these limitations, in this paper, we provide an engineering-oriented solution of precise hovering based on visual servoing and fractional order proportional-integral-derivative (PID) controller without GPS information. First, the mathematical speed model of a quadcopter is obtained by step response experiments. Then, fractional order PID controller algorithm is designed to improve its hovering accuracy and robustness. In our work, the position reference is extracted with an on-board color based target recognition algorithm instead of GPS. Both simulation and field experimental results demonstrate that the proposed scheme can achieve a better performance in terms of hovering accuracy and robustness to disturbances. In particular, for the first time, we show that, the sampling period can be bigger than usual when using fractional order control algorithm, relaxing costly hardware requirement for fast real-time vision-based feedback.**

Index Terms— **Unmanned aerial vehicles (UAVs), Visual Servoing Controller, Fractional order proportional-integral-derivative(FOPID), Large sampling period**

I. Introduction

Autonomous flight of Unmanned Aerial Vehicle (UAV) is needed in many situations, like urban search and rescue missions [1], surveillance [2], formation flight [3], smart city [4] and biological chemical agent detection [5]. Takeoff, hovering, navigation and landing are four basic processes for autonomous flight of UAVs. Among them, hovering is especially important and complex when UAV is used to do some manipulations [6], air-to-air refuelling [7] or recharging [8], spraying [9] and precise inspection [10]. As to micro UAVs, limited payload can be carried onboard so multiple-functional sensors are more preferred, such as GPS, IMU, compass and etc. However, GPS is unavailable in some areas or in certain indoor environment. Furthermore, the precision of civil GPS may not satisfy the requirement of some precise positioning tasks. Therefore, precision hovering without GPS support is challenging in practice. In this case, camera-based vision is a practical sensor which can provide more information for precision hovering and landing.

Previous works on visually guided takeoff, hovering and landing of UAVs were mainly focused on the landing problem due to high risks and reliability. Many previous works related to vision-based landing have done intensive research on real-time vision-based sensing and control. [11] presented an exhaustive survey of some landing algorithms and systems, both unmanned rotory and fixed-wing aircrafts using visual sensors are included. In this survey, the main research groups involved in the development of vision-based autonomous landing systems were presented, and then it discussed the details of each algorithm and system in different categories. [12] presents a review of landing techniques ranging from GPS based landing to vision-based landing techniques and from basic nonlinear control method to intelligent, hybrid and robust ones.

For vision-guided hovering of UAVs, [13] developed an on-board FPGA vision platform to eliminate drift while hovering, using Harris feature detection and template matching method. [14] focused on the higher level system architecture and key supporting algorithms for the guidance and control of an HMAV in unknown, GPS-denied environments. The Shi-Tomasi "good features to track" detector was adopted to find points with large eigenvalues of the image gradient in multiple directions, and then a mask is erected in the vicinity of existing features, within which new features are not detected. Finally, the pyramidal Lucas-Kanade optical flow calculator compares pixel patches in successive images in an attempt to accurately maintain the locations of features detected in previous images. However, the image processing is off-board. [15] integrated a tiny single camera with Inertial Measurement Units to hover the UAV over specified markers as an external reference for pose estimation where PID controllers are used to control the pose. But, the images are processed on a ground station PC. [16] modified appropriately the sliding window over which the Multi-State Constrained Kalman Filter (MSC-KF) operates, so as to perform 3D localization robustly both under hovering and generic motion conditions, without the need of building a map of the environment. Experiment results are carried out on a low-cost AR.DRONE quadrotor, and images and inertial measurements were streamed through the wireless module of the quadrotor UAV and processed in real-time on a ground station computer. [17] proposed a PD controller which uses optical flow to obtain position and velocity feedback, fusing with IMU data without any support of GPS data. A micro wireless camera points directly down and transmits video to the ground station at the frequency of 2.4 GHz, and then

Bo Shang is with College of Information Science and Engineering, Northeastern University, Shenyang, China. Email:(cnpcshangbo@gmail.com). Jianxin Liu is with School of Mechanical Engineering, Xihua University, Chengdu, Sichuan, China. Email:(jianxin2013liu@outlook.com). Tiebiao Zhao and YangQuan Chen are with Mechatronics, Embedded Systems and Automation Lab, University of California, Merced, CA 95343, USA. Email:(tzhao3, ychen53@ucmerced.edu).

*Corresponding author. Email: cnpcshangbo@gmail.com, T: +1(209)658-4779, W: http://mechatronics.ucmerced.edu

978-1-4673-9333-1/16/$31.00 ©2016 IEEE

optic flow calculation and data fusing are implemented on a ground PC. The indoor experimental hovering precision is shown to be around 25cm.

So far, it is quite clear that the on-board image processing and hovering control still needs extensive developments considering long range and long time hovering beyond the constraint of WiFi distance. Furthermore, hovering precision and anti-disturbance such as gust in outdoor environment need more work.

In this paper, for the first time, we show that, the sampling period can be bigger than usual when using fractional order control algorithm, relaxing costly hardware requirement for fast real-time vision-based feedback.

The paper is organized as follows. The preliminaries of fractional calculus, specifications for fractional order controller design are discussed in Sec. I; and in Sec. II system architecture and model identification of the UAV are presented. In Sec. III, fractional order controller synthesis steps are presented. Simulation results are presented in Sec. IV. Experimental results are presented and discussed in Sec. V. Section VI concludes with paper with an outline of further research efforts we planned.

A. Definition of fractional differentiation

Fractional differentiation was proposed in the seventeenth century and developed mainly in the nineteenth [18]. Many definitions of fractional differentiation have been proposed and the most popular definitions among them are: the Riemann-Liouville(R-L) definition, the Grünwald-Letnikov (G-L) definition [19] and the Caputo definition [20].

1. The Riemann-Liouville(R-L) definition

$$_aD_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{(\alpha-n+1)}} \mathrm{d}\tau, \tag{1}$$

where, $f(t)$ is a function assumed to be sufficiently smooth and locally integrable.

2. The Grünwald-Letnikov(G-L) definition

$$_aD_t^{\alpha} f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{j=0}^{[\frac{t-a}{h}]} (-1)^{j}\,{}^{\alpha}C_j f(t-jh), \tag{2}$$

where, ${}^{\alpha}C_j$ is the binomial coefficient.

3. The Caputo definition

$$_aD_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^n(\tau)}{(t-\tau)^{(\alpha-n+1)}} \mathrm{d}\tau. \tag{3}$$

Among these three definitions, the most commonly used is the Riemann-Liouville (R-L) definition. The R-L definition and Caputo definition are almost the same except for some initial value settings. The Laplace transform under zero initial conditions can be represented as:

$$L\{_aD_t^{\pm\alpha} f(t)\} = s^{\pm\alpha} F(s), \tag{4}$$

where, $0 < \alpha < 1$.

B. Specifications for controller design

Considering the case of a plant $P(s)$ with a fractional order controller $C(s)$, the open loop transfer function (TF) is $G(s) = C(s)P(s)$. There are three specifications for controller design, given the expected gain crossover frequency ω_c and phase margin ϕ_m, as following[21], [22]:

1). Phase margin specification

$$Arg[G(j\omega_c)] = Arg[C(j\omega_c)P(j\omega_c)] = -\pi + \phi_m. \tag{5}$$

2). Robustness to variation in the gain of plant

$$\frac{\mathrm{d}(Arg[C(j\omega)P(j\omega)])}{\mathrm{d}\omega}|_{\omega=\omega_c} = 0, \tag{6}$$

with the condition that the phase derivative w.r.t. the frequency is zero, which means that the system is more robust to gain changes and the overshoots of the responses are almost the same.

3. Gain crossover frequency specification

$$|G(j\omega_c)|_{\mathrm{dB}} = |C(j\omega_c)P(j\omega_c)|_{\mathrm{dB}} = 0. \tag{7}$$

II. System architecture and model identification of the UAV

A. System architecture

The platform we used is based on an internal report [23]. The UAV with real-time vision-based hovering capability mainly includes the following parts: quadcopter body, Pixhawk autopilot system [24], radio receiver, battery, camera and companion computer, which are shown in Fig. 1. The quadcopter body and Pixhawk autopilot system can form a flying system that can be manually operated by remote control. Odroid XU3 [25] is used as the companion computer, which is in charge of on-board image processing and high-level control. Odroid is connected to Pixhawk with an FTDI cable which can convert from USB to serial port. OpenCV is used to get video frames from camera and detects the red target circle we want to track. DroneKit-Python 2.0 is used to send speed command to the Pixhawk autopilot system. The software architecture in the Odroid is shown in Fig. 2. In this way, we can give feedback information to the drone according to the visual information. A laptop computer is connected to the Odroid by WiFi to start and stop the script as well as monitoring the flight status including battery voltage level.

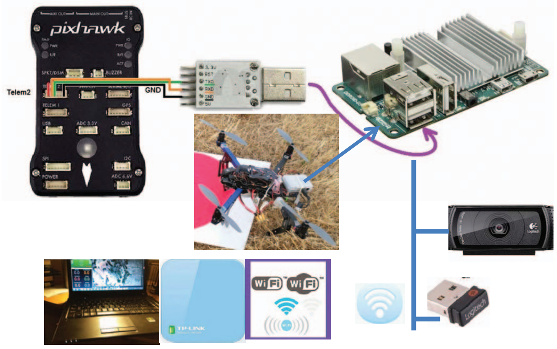

Fig. 1. Real-time vision-based hovering system architecture

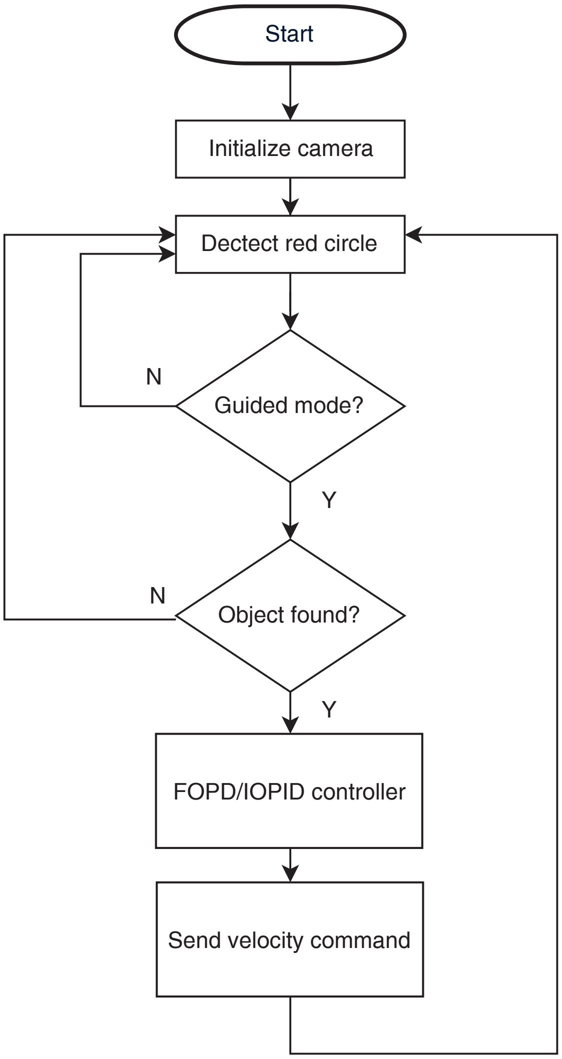

Fig. 2. Software architecture block-diagram

B. Model identification of the UAV

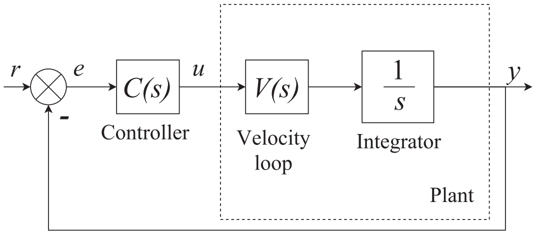

Fig. 3. Structure of the overall control system.

Figure 3 shows the structure of control system, where the plant and controller are respectively described by transfer functions $P(s)$ and $C(s)$. Also, $r(t)$, $u(t)$, and $y(t)$ respectively denote the reference input, the control signal, and the system output. Moreover, $e(t)=r(t)-y(t)$ is the error signal in the aforementioned control system.

Given step speed signal to the UAV, we can get the speed response of the quadcopter. Using the System Identification Toolbox in Matlab, the model can be obtained as shown in Fig. 4.

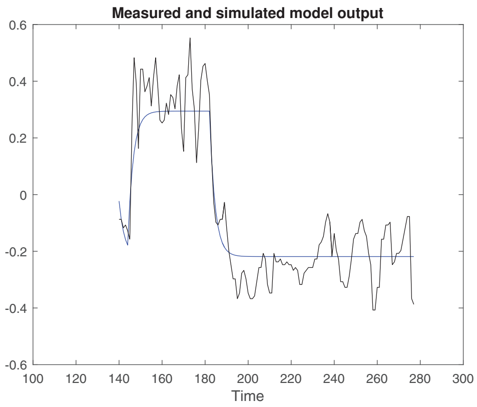

Fig. 4. Speed model system identification.

The identified speed model transfer function is:

$$V(s)=\frac{1.0263}{1+0.71s}. \tag{8}$$

Therefore, the plant transfer function is:

$$P(s)=\frac{1.0263}{0.71s+1}\frac{1}{s}, \tag{9}$$

which is a second order system.

III. Design of the hovering controllers

A. Design of FOPD hovering controller

1) Frequency domain model of the FOPD controller: The transfer function of FOPD (fractional order proportional derivative) controller is:

$$C(s)=K_p(1+\frac{K_i}{s^{\lambda}}). \tag{10}$$

By substituting s with $j\omega$, PD^{λ} controller can be described as:

$$C(j\omega)=K_p[(1+K_i\omega^{\lambda}\cos\frac{\pi\lambda}{2})+jK_i\omega^{\lambda}\sin\frac{\pi\lambda}{2}]. \tag{11}$$

The phase and gain are as follows:

$$Arg[C(j\omega)]=\arctan\frac{K_i\omega^{\lambda}\sin\frac{\pi\lambda}{2}}{1+K_i\omega^{\lambda}\cos\frac{\pi\lambda}{2}}, \tag{12}$$

$$|C(j\omega)|=K_pJ(\omega), \tag{13}$$

where

$$J(\omega)=\sqrt{(1+K_i\omega^{\lambda}\cos\frac{\pi\lambda}{2})^2+(K_i\omega^{\lambda}\sin\frac{\pi\lambda}{2})^2}. \tag{14}$$

2) Frequency domain model of the plant: The transfer function of the plant is:

$$P(s) = \frac{k}{s(\tau s + 1)}. \tag{15}$$

After substituting s with $j\omega$, the plant can be described as:

$$P(j\omega) = -\frac{k(\tau\omega + j)}{\omega(\tau^2\omega^2 + 1)}. \tag{16}$$

The phase and gain are as follows:

$$Arg[P(j\omega)] = \arctan\frac{1}{\tau\omega}, \tag{17}$$

$$|P(j\omega)| = \frac{k}{\omega(\tau^2\omega^2 + 1)}\sqrt{\tau^2\omega^2 + 1}. \tag{18}$$

3) Frequency domain response of the open loop transfer function: The open loop transfer function is $G(s) = C(s)P(s)$. From (12) and (17), the phase of $G(s)$ is as follows:

$$Arg[G(j\omega)] = arctan\frac{K_i\omega^\lambda \sin\frac{\pi\lambda}{2}}{1 + K_i\omega^\lambda \cos\frac{\pi\lambda}{2}} + arctan\frac{1}{\tau\omega}. \tag{19}$$

The gain of $G(s)$ is:

$$|G(j\omega)| = K_pJ(\omega)\frac{k}{\omega(\tau^2\omega^2 + 1)}\sqrt{\tau^2\omega^2 + 1}. \tag{20}$$

4) Tuning of PD^λ controller: According to the performance index specifications in (5) and (6), the parameters of K_i and λ should satisfy the equation (21) and (22) simultaneously when ω is set to the gain crossover frequency ω_c.

$$K_i = \frac{\tan(\arctan\frac{1}{\tau\omega} - \phi_m)}{-\omega^\lambda \sin\frac{\pi\lambda}{2} - \omega^\lambda \cos\frac{\pi\lambda}{2}\tan(\arctan\frac{1}{\tau\omega} - \phi_m)}. \tag{21}$$

$$A_1K_i^2 + A_2K_i + A_3 = 0, \tag{22}$$

where,

$$A_1 = \tau\omega^{2\lambda}, \tag{23}$$

$$A_2 = 2\tau\omega^\lambda \cos\frac{\pi\lambda}{2} - \sin\frac{\pi\lambda}{2}\omega^{\lambda-1}\lambda(1 + \lambda^2\omega^2), \tag{24}$$

$$A_3 = \tau. \tag{25}$$

Therefore, the parameter K_i should satisfy the equation (26),

$$K_i = \frac{-A_2 \pm \sqrt{A_2^2 - 4A_1A_3}}{2A_1}. \tag{26}$$

Furthermore, the parameter of K_p should satisfy the equation (27),

$$|G(j\omega)| = K_pJ(\omega)\frac{k}{\omega\sqrt{\tau^2\omega^2 + 1}} = 1. \tag{27}$$

B. Design of the PID controller

There are many methods to tune the parameters of PID controller, we chose the function of `pidtune` in Matlab [26], which can achieve a good balance between performance and robustness [27]. We have got the following PID parameters for the plant transfer function in Eq. (9):

$$K_p = 2.89, \tag{28}$$

$$K_i = 1.04, \tag{29}$$

$$K_d = 1.79. \tag{30}$$

Therefore, we have got the IOPID (integer-order proportional integral derivative) controller transfer function:

$$C_{IOPID}(s) = 2.89 + \frac{1.04}{s} + 1.79s. \tag{31}$$

The Bode plot of the above tuned PID controller is shown in Fig. 5.

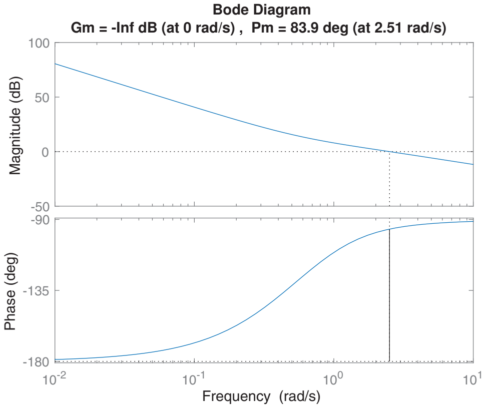

Fig. 5. Bode plot of IOPID controller tuned by Matlab

From the Bode plot, we can get to know the $\omega_c = 2.51$ rad/sec. and $\phi_m = 83.9$ deg, which will be used to design the FOPID controller for fair comparison.

IV. Simulation Results

A. FOPD controller parameter calculation and verification

We set the same ω_c and ϕ_m for FOPD controller. λ is set to be between 0 and 2. Then we could use the graphical method to solve the equations (21) and (26). We have got the solution:

$$\lambda = 0.9694, \tag{32}$$

$$K_i = 0.6192. \tag{33}$$

By using (27) we can calculate that $K_p = 2.6992$. Therefore the transfer function of the fractional order controller is:

$$C_{FOPD}(s) = 0.6192s^{0.9694} + 2.6992. \tag{34}$$

Let us compare the Bode plots of FOPD and IOPID controllers to see if the solution is correct. As shown in Fig. 6, the FOPD controller has the correct ω_c and ϕ_m as we designed.

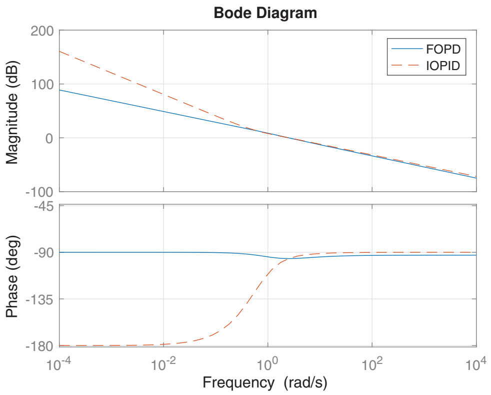

Fig. 6. FOPD controller and IOPID controller Bode plots

B. Step response simulation of FOPD

We used Matlab Simulink to simulate the FOPD control system and the IOPID control system. As shown in Fig. 7, the step response has been improved in its rising time and overshoot when the FOPD is adopted.

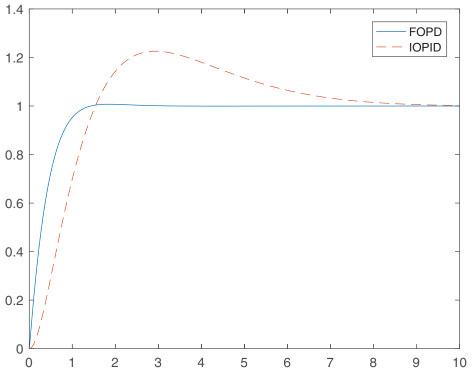

Fig. 7. Comparison of FOPD and IOPID step responses

C. Approximation accuracy

Considering that we need to implement the controllers, the FOPD controller needs to be converted to a high-order integer transfer function. For a given fractional order transfer function, different approximation methods can be used, such as ORA (Oustaloup Recursive Approximation) [28] and IRID (Impulse Response Invariant Discretion) [29]. The IRID method can guarantee the approximation accuracies consistently high in both time and frequency domain when the sampling period is small enough. However, in our real-time vision-based feedback control system, due to low cost requirement, the hardware performance is limited resulting the sampling period as big as 0.58 sec.. IRID is obviously more sensitive to the large sampling period used. Therefore, the ORA method is used to implement the fractional order controller transfer function in this work. We used the fractional transfer function (34) as the input of the ORA method. Since the $\omega_c = 2.51$ rad/sec., we have selected the frequency range from $\frac{\omega_c}{1000}$ to $100\omega_c$, then we obtained the result of the approximated integer transfer function (continuous time domain). The `c2d` command [30] in Matlab Control Systems Toolbox is used with the `Tustin` parameter to convert the integer transfer function to discrete-time domain with the sampling period of 0.58 sec.. Matlab Simulink z domain transfer function block is used to simulate the FOPD control system. For the IOPID controller, we used discrete PID controller block and the `pidtune` tuned parameters of the continuous IOPID controller. Figure 8 shows the FOPD controller has a better performance than the IOPID controller after discretization. This is an important desirable advantage we discovered in this work for the first time.

D. Resilience against large sampling periods

As a summary for the previous content, we have got transfer functions of the plant and two controllers. The plant transfer function is:

$$P(s) = \frac{1.0263}{0.71s+1}\frac{1}{s}. \tag{35}$$

The IOPID (integer-order proportional integral derivative) controller transfer function is:

$$C_{IOPID}(s) = 2.89 + \frac{1.04}{s} + 1.79s. \tag{36}$$

The FOPD (fractional-order proportional derivative) controller transfer function is:

$$C_{FOPD}(s) = 0.6192s^{0.9694} + 2.6992. \tag{37}$$

Then we used ORA to approximate the FOPD controller. Since we are using digital control system, we need to convert the continuous controllers into discrete controllers. The sampling time is an important parameter for this discretization. We tried to investigate the robustness against large sampling periods by simulation with different sampling periods. We tried four different sampling times (T_s=0.58 sec., 0.33 sec., 0.29 sec. and 0.058 sec.) when using `c2d` command in Matlab.

Figures 8, 9, 10 and 11 show that FOPD controller is more resilient again large sampling period than IOPID controller.

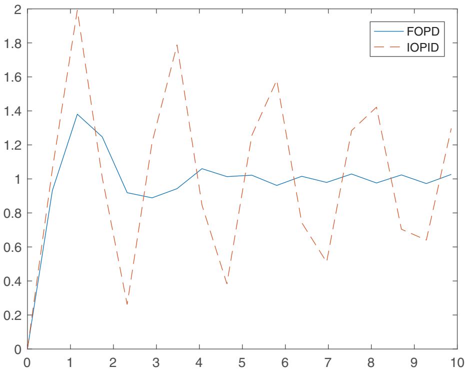

Fig. 8. Comparison of FOPD and IOPID step responses (T_s=0.58 sec.)

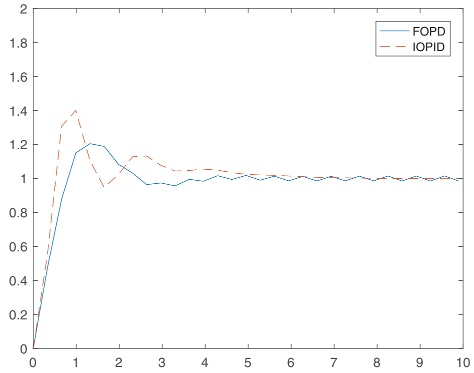

Fig. 9. Comparison of FOPD and IOPID step responses (T_s=0.33 sec.)

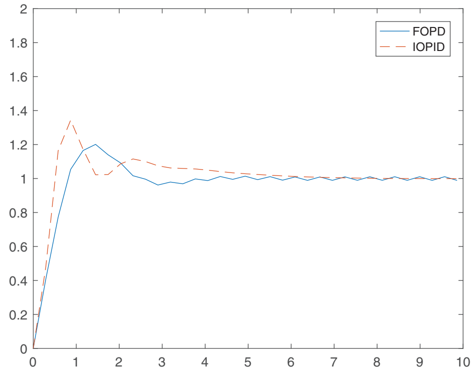

Fig. 10. Comparison of FOPD and IOPID step responses (T_s=0.29 sec.)

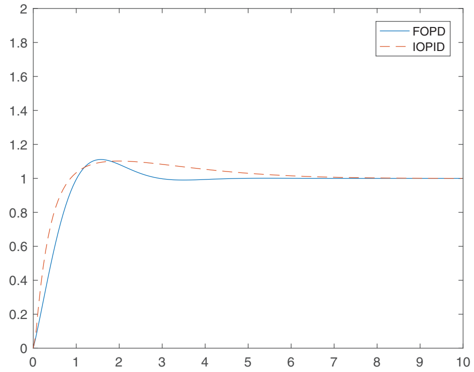

Fig. 11. Comparison of FOPD and IOPID step responses (T_s=0.058 sec.)

V. Experimental results

After the controllers are verified by simulation study, we performed real flight tests. We tested the FOPD controller and IOPID controller by implementing two tests in the following which we can ensure that they are conducted in similar wind speed. For each flight, we first flew the drone to 10 meters high manually and then switch to guided mode by RC. In that mode, the drone would be controlled with real-time vision feedback. We flew one minute for each controller and collected the position data. Then we plotted those data in Fig. 12 and Fig. 13. The dots from FOPD controller are within a circle whose radius is 58 pixels, while for IOPID the radius of the circle is 68 pixels. In this way, we can see the FOPD controller has a nearly 20% better performance compared to IOPID controller.

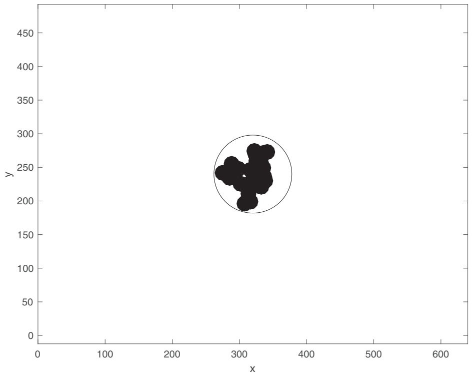

Fig. 12. Experiment results for FOPD controller.

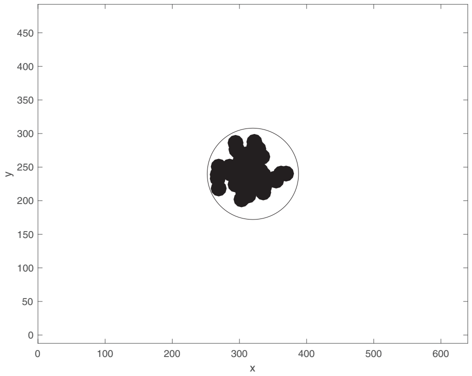

Fig. 13. Experiment results for IOPID controller.

VI. Conclusion

This paper designed a fractional order proportional derivative (FOPD) controller for quadrotor UAV visual position servoing. An integer order PID (IOPID) controller is designed for fair comparison with the same phase margin and gain crossover frequency. Step responses of the two controllers are simulated, which show the fastness of the FOPD controller. The robustness of the FOPD controller is demonstrated by drawing the Bode plot to compare the phase margin and gain margin. We also simulated the situation that the model of the plant changing in gain K_s and time constant τ, showing that the FOPD controller has a better robustness against the model changes. After the simulation, the designed FOPD controller was converted to integer transfer function and implemented by a discrete-time controller. Real flight tests proved that the conclusion we drew in the simulation. In the future, we are going to get more experiment results from different platforms and we'll try to get the drone start-up smoother by fractional order method to prevent the target from disappearing from view.

Acknowledgment

This work was supported in part by the research project granted by Xihua University and China Scholarship Council (CSC). The authors Bo Shang and Jianxin Liu contributed equally to this work.

References

[1] Teodor Tomić, Korbinian Schmid, Philipp Lutz, Andreas Dömel, Michael Kassecker, Elmar Mair, Iris Lynne Grixa, Felix Ruess, Michael Suppa, and Darius Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

[2] Jerry Burman, Joao Hespanha, Upamanyu Madhow, Jason Isaacs, Sriram Venkateswaran, and Tien Pham, "Autonomous UAV persistent surveillance using bio-inspired strategies," in *Proceedings of SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2012, pp. 838917–838917.

[3] Zouhair Mahboubi, Zico Kolter, Tao Wang, Geoffrey Bower, and Andrew Y Ng, "Camera based localization for autonomous UAV formation flight," in *Proceedings of the AIAA Infotech Conference*, 2011.

[4] Gabriele Ermacora, Antonio Toma, Stefano Rosa, Basilio Bona, Marcello Chiaberge, Mario Silvagni, Marco Gaspardone, and Roberto Antonini, "A cloud based service for management and planning of autonomous UAV missions in smart city scenarios," in *Proceedings of Modelling and Simulation for Autonomous Systems: First International Workshop, MESAS 2014, Rome, Italy, May 5-6, 2014, Revised Selected Papers*. Springer, 2014, vol. 8906, p. 20.

[5] Chen Friedman, Inderjit Chopra, Svetlana Potyagaylo, Omri Rand, and Y Kanza, "Towards obstacle avoidance and autonomous UAV operation," in *Proc. of the AHS International Meeting on Advanced Rotorcraft Technology and Safety Operations (HeliJapan), Phoenix, AZ. AHS International*. Citeseer, 2011.

[6] Matko Orsag, Christopher Korpela, and Paul Oh, "Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 227–240, 2013.

[7] Carlos Martinez, Tom Richardson, and Pascual Campoy, "Towards autonomous air-to-air refuelling for UAVs using visual information," in *Proc. of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 5756–5762.

[8] Elizabeth Basha, Michael Eiskamp, Jennifer Johnson, and Carrick Detweiler, "UAV recharging opportunities and policies for sensor networks," *International Journal of Distributed Sensor Networks*, vol. 501, pp. 158–168, 2015.

[9] Bruno S Faiçal, Fausto G Costa, Gustavo Pessin, Jó Ueyama, Heitor Freitas, Alexandre Colombo, Pedro H Fini, Leandro Villas, Fernando S Osório, Patrícia A Vargas, et al., "The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides," *Journal of Systems Architecture*, vol. 60, no. 4, pp. 393–404, 2014.

[10] Inkyu Sa, Silvio Hrabar, and Peter Corke, "Inspection of pole-like structures using a vision-controlled VTOL UAV and shared autonomy," in *Proc. of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014),*. IEEE, 2014, pp. 4819–4826.

[11] Weiwei Kong, Dianle Zhou, Daibing Zhang, and Jianwei Zhang, "Vision-based autonomous landing system for unmanned aerial vehicle: A survey," in *Proc. of the 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*. IEEE, 2014, pp. 1–8.

[12] Anjali Gautam, P. B. Sujit, and Srikanth Saripalli, "A survey of autonomous landing techniques for UAVs," in *Proc. of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 1210–1218.

[13] Spencer G Fowers, Dah-Jye Lee, Beau J Tippetts, Kirt D Lillywhite, Aaron W Dennis, and James K Archibald, "Vision aided stabilization and the development of a quad-rotor micro UAV," in *Proc. of the International Symposium on Computational Intelligence in Robotics and Automation, 2007. CIRA 2007.* Ieee, 2007, pp. 143–148.

[14] Spencer Ahrens, Daniel Levine, Gregory Andrews, and Jonathan P How, "Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments," in *Proc. of the IEEE International Conference on Robotics and Automation, 2009. ICRA'09.* IEEE, 2009, pp. 2643–2648.

[15] Tianguang Zhang, Ye Kang, Markus Achtelik, Kolja Kühnlenz, and Martin Buss, "Autonomous hovering of a vision/IMU guided quadrotor," in *Proc. of the International Conference on Mechatronics and Automation, 2009. ICMA 2009.* IEEE, 2009, pp. 2870–2875.

[16] Dimitrios G Kottas, Kejian J Wu, Stergios Roumeliotis, et al., "Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems," in *Proc. of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 3172–3179.

[17] Xian Bin, Liu Yang, Zhang Xu, Cao Meihui, and Wang Fu, "Hovering control of a nano quadrotor unmanned aerial vehicle using optical flow," in *Proc. of the 2014 33rd Chinese Control Conference (CCC)*. IEEE, 2014, pp. 8259–8264.

[18] Igor Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, vol. 198, Academic press, 1998.

[19] Kenneth S Miller, "Derivatives of noninteger order," *Mathematics Magazine*, pp. 183–192, 1995.

[20] M Caputo, "Linear models of dissipation whose Q is almost frequency independent," *Annals of Geophysics*, vol. 19, no. 4, pp. 383–393, 1966.

[21] Ying Luo and YangQuan Chen, "Fractional order [proportional derivative] controller for a class of fractional order systems," *Automatica*, vol. 45, no. 10, pp. 2446–2450, 2009.

[22] HongSheng Li, Ying Luo, and YangQuan Chen, "A fractional order proportional and derivative (FOPD) motion controller: tuning rule and experiments," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 516–520, 2010.

[23] Jianxin Liu, Bo Shang, Tiebiao Zhao, and YangQuan Chen, "More robust hovering of a quad-rotor UAV using real-time vision-based fractional order visual servoing controller," Tech. Rep., MESA LAB, UC Merced, March 2016.

[24] Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[25] "ODROID-XU3," `http://www.hardkernel.com/`, Accessed: 2016-2-20.

[26] "PID tuning algorithm for linear plant model," `http://www.mathworks.com/help/control/ref/pidtune.html`, Accessed: 2016-5-5.

[27] Karl Johan Åström and Tore Hägglund, *Advanced PID control*, ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.

[28] "Oustaloup-Recursive-Approximation for fractional order differentiators," `http://www.mathworks.com/matlabcentral/fileexchange/3802-oustaloup-recursive-approximation-for-fractional-order-differentiators`, Accessed: 2016-2-23.

[29] "Impulse response invariant discretization of fractional order integrators/differentiators," `http://www.mathworks.com/matlabcentral/fileexchange/21342-impulse-response-invariant-discretization-of-fractional-order-integrators-differentiators`, Accessed: 2016-2-23.

[30] "Convert model from continuous to discrete time," `http://www.mathworks.com/help/control/ref/c2d.html`, Accessed: 2016-5-5.