Design of a Flight Stabilizer System for a Small Fixed Wing Unmanned Aerial Vehicle using System Identification

Halim KORKMAZ[1, a], Onur Baki ERTİN[1,b], Coşku KASNAKOĞLU[1,c], Ünver KAYNAK[2,d]

***Abstract*—Flight stabilizers for unmanned aerial vehicles (UAVs) provide level flight and make the UAVs more reliable and operational. In this paper a system identification process is applied to flight data logs obtained from flight simulations and a stabilizer system is designed using the dynamical model obtained from system identification. Hardware-in-the-loop testing was also performed on the flight stabilizer system designed is also tested using a custom built moving platform in the pitch and roll axis with actual flight sensors.**

***Index Terms*—Unmanned aerial vehicle, UAV, flight stabilizer, system identification, test platform, hardware-in-the-loop.**

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are one of the popular topics in aeronautics, which have recently gained great importance in both military and civil applications. They are becoming more reachable because of the increasing efficiency and reduced cost of the flight components such as batteries, electric motors, and controlling boards. [1-2].

Autopilot systems are the major area of design for UAVs. These systems perform autonomous flights for a desired aerial vehicle and a flight mission can be done without human input. Nowadays a UAV can also land and take off automatically via an autopilot system [3].

Flight stabilizer is an important part of an autopilot system. It corrects flight angle errors and allows the plane perform a level flight. When a disturbance is applied to the aerial vehicle, flight stabilizer counteracts this effect and provides a smoother flight pattern compared to manual flight.

In this paper, we outline a system identification based approach to building a flight stabilizer for a small fixed winged UAV. First, flight data logs are collected for a popular RC plane, namely PT-60, using the Xplane flight simulator program. The Xplane software is very useful for flight controller design for all kind of aircrafts. It is easy to manipulate, a very realistic flight and also allows for the alteration of various weather conditions. After logging the flight data in Xplane, system identification is applied for various input and output parameters of the flight data logs using MATLAB [4]. Then a flight stabilizer control system is designed using standard tuning methods on the mathematical model identified for the RC plane. When the design procedure is completed, an hardware-in-the-loop test of the entire system is performed using a custom built moving hardware platform and the results were found to be satisfactory. The succeeding sections of the paper describe these steps in detail.

II. DESIGN STAGES

A. Collecting flight data

For this stage, Xplane flight simulation program is used for data logging in hardware in the loop mode (HIL). Firstly, Xplane is set for data logging process. Then a PT-60 RC plane is flown manually for a while. During flight, the aircraft is forced for sharp maneuvers and bad weather conditions are applied to the flight environment for more reliable and realistic system identification results. When the flight is completed, pulse width modulation (PWM) signal values for control surfaces and attitude angles are recorded to a log file as different columns. PWM signal values are scaled from $(1000, 2000)$ to $(-1, 1)$ interval automatically by Xplane and attitude angles (roll, pitch, and yaw) are recorded in degrees. PWM signals are recorded for aileron, elevator, rudder control surfaces and throttle. The PT-60 test aircraft used for this process is shown in Fig. 1. Overall data logging setup with hardware in loop (HIL) system is shown in Fig. 2. together with flight control units.

Manuscript received September 20, 2012. This work is supported by the Scientific & Technological Research Council of Turkey (TUBITAK) through the 1507 program under project number 7110829.

Halim Korkmaz is with the Department of Electrical and Electronics Engineering, TOBB ETU, 06560, Ankara, TURKEY (e-mail: hkorkmaz@etu.edu.tr).

Onur Baki Ertin is with the Department of Electrical and Electronics Engineering, TOBB ETU, 06560, Ankara, TURKEY (e-mail: bertin@etu.edu.tr).

Coşku Kasnakoğlu is with the Department of Electrical and Electronics Engineering, TOBB ETU, 06560, Ankara, TURKEY (e-mail: kasnakoglu@etu.edu.tr).

Ünver Kaynak is with the Department of Mechanical Engineering, TOBB ETU, 06560, Ankara, TURKEY (e-mail: ukaynak@etu.edu.tr).

Figure 1: PT60 RC aircraft

Figure 2: Overall flight data logging setup

B. *System identification*

After collecting flight data, system identification is applied around the zero point separately for specific input and output signals by utilizing MATLAB [5-9]. The input variables are chosen as the scaled aileron and elevator PWM signals and the output variables are chosen as the roll and pitch angles in degrees respectively. Rudder input scaled PWM signal and yaw output angle were left out since they are of less importance for a flight stabilizer and including them would increase the system complexity unnecessarily. Input scaled PWM signals are shown in Fig. 3.

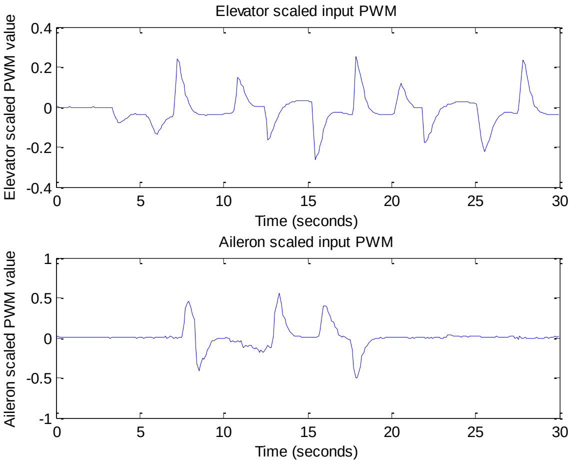

Figure 3: Input scaled PWM signals

Output signals for input scaled PWM signals are shown in Fig. 4.

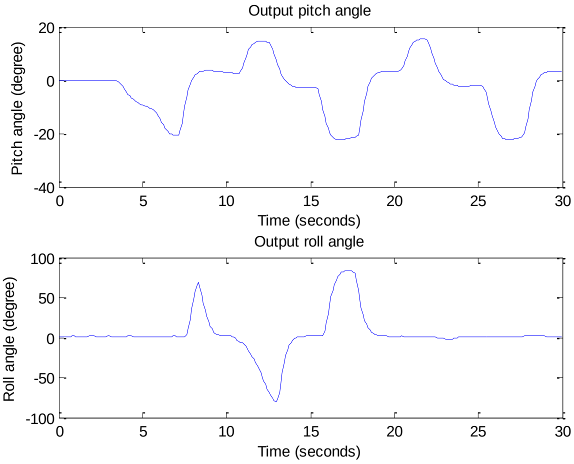

Figure 4: Output attitude signals

Transfer functions for the roll model R(s) and the pitch model P(s) which were obtained by processing the input-output data through MATLAB System Identification Toolbox are given below.

$$R(s) = \frac{9.629e04\ s + 1105}{s^3 + 363.1\ s^2 + 0.1315\ s + 9.496}$$

$$P(s) = \frac{7.035\ s^2 + 2467s + 659.7}{s^3 + 20.03\ s^2 + 4.079\ s + 5.087}$$

R(s) has 3 poles and 1 zero, P(s) has 3 poles and 2 zeros. The structures of the models obtained are quite similar to each other. Estimated output signals are obtained by using input signals in Fig. 3. and R(s), P(s) transfer functions. Estimated output pitch and roll angles are shown in Fig. 5.

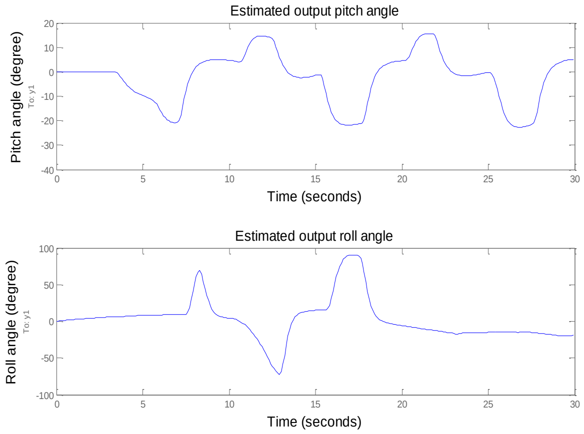

Figure 5: Estimated output pitch and roll angles

Note that these signals are very similar to the output signals shown in Fig. 4. This similarity confirms that system identification process is completed correctly and the models obtained are capable of capturing the flight dynamics satisfactorily.

C. *Designing a PID controller for the identified system*

At this stage a PID controller is designed for each identified system separately. MATLAB SISO Design Tool is utilized for compensator design. According to the SISO Design Tool, the optimum compensator solution is found to be a PI controller for both the roll model R(s) and the pitch model P(s).

For roll model R(s), the PI compensator obtained is shown below.

$$C(s)_{roll} = 0.47825\ x\ \frac{1+0.98s}{s}$$

Proportional and Integral constant for $C(s)_{roll}$ are 0.468685 and 0.47825 respectively. For roll model P(s), the PI compensator obtained is shown below.

$$C(s)_{pitch} = 0.3059\ x\ \frac{1+1.7s}{s}$$

Proportional and Integral constant for $C(s)_{pitch}$ are 0.52003 and 0.3059 respectively.

Step responses for $C(s)_{roll}$ and $C(s)_{pitch}$ are shown in Fig. 6. and Fig.7. respectively. Although $C(s)_{roll}$ gives slower reaction than $C(s)_{pitch}$, both compensators have enough and realistic responses for step inputs. This situation will be clearer by observing the flight test output signals in the later sections.

At this point, a basic flight stabilizer system has been designed and is ready to be used in flight tests.

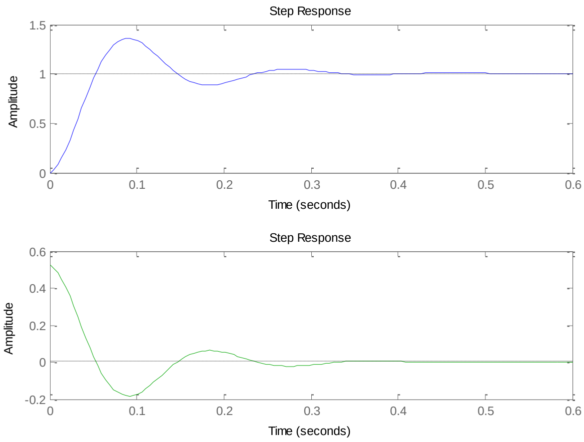

Figure 6: Step response for $C(s)_{pitch}$

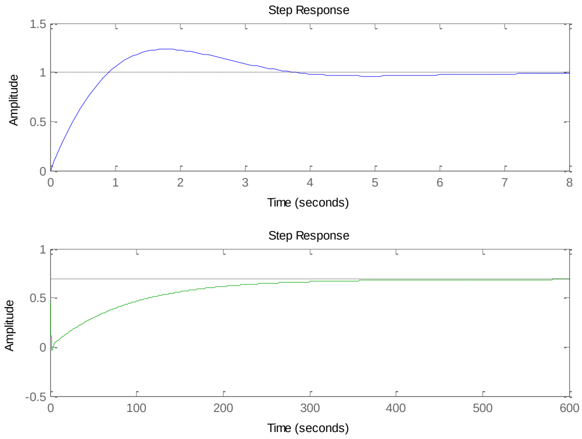

Figure 7: Step response for $C(s)_{roll}$

III. TESTS

Designed system is tested with a moving hardware platform in the pitch and roll axis with actual flight sensors to form a complete testing environment [10-11].

Moving hardware platform includes two servos which provide pitch and roll movements. Whole hardware platform stabilize system is controlled by Ardupilot Mega microcontroller board.

Ardupilot Mega has its own IMU and magnetometer sensors. It also has open source libraries so it gives a chance to developers to embed and test custom code. Xplane simulation program sends attitude data to Ardupilot Mega and then Ardupilot Mega calculates stabilizing responses and sends them to Xplane back through a C# application. Ardupilot Mega is commanded by a 2.4Ghz RC remote controller. Xplane also sends attitude data to Arduino Mega through another C# application. Servos are positioned to pitch and roll angles by Arduino Mega. Ardupilot Mega is attached to the surface of the platform. So, actual flight sensors are used for the whole controlling mechanism. Moving hardware platform system connections are shown in Fig. 8.

The moving hardware platform has a limited operating range, because all connected servos have angle position limit and they cannot go further. Therefore the moving hardware platform cannot do extreme movements such as a barrel roll or a pitch circle. For this reason the flight tests were not pushed to such limits.

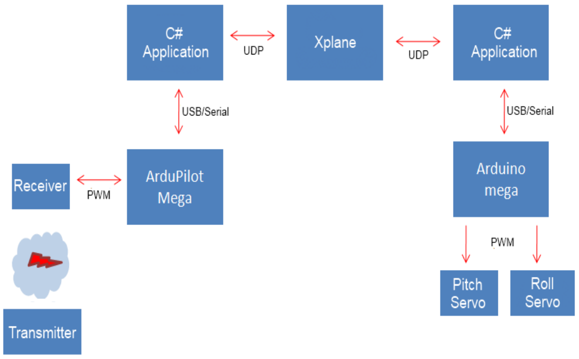

Figure 8: Moving hardware platform system connections

The entire moving hardware platform is shown in Fig. 9.

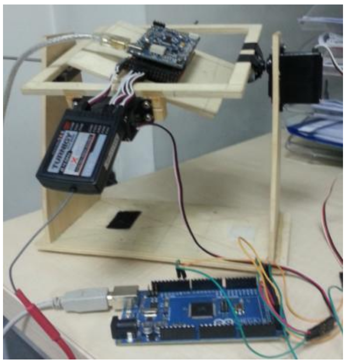

Figure 9: Moving hardware platform

Custom PID controllers were designed and embedded to Ardupilot Mega for pitch and roll axis. PID parameters are taken from the previous section which includes PID controller design for the identified system.

IV. RESULTS

PT-60 RC aircraft is simulated on Xplane with the moving hardware platform and the flight results are examined. During the simulation process, PID coefficients are used which were designed separately for the transfer functions identified for roll and pitch. Also, the error signal (reference angle minus the pitch/roll angle) is scaled and preprocessed before being fed to the PID controller. A block diagram of the flight stabilizer feedback controlling scheme is shown in Fig. 10. Reference inputs are assumed to be zero because stabilizing means holding the aircraft at zero pitch and zero roll degrees for level flight.

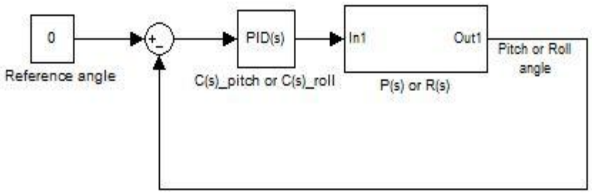

Figure 10: Flight stabilizer feedback controlling scheme

Data logging process is enabled for the newly obtained PID coefficients during flight tests with moving hardware platform again. According to the data logs, input scaled PWM signals are shown in Fig. 11. and output signals for these input scaled PWM signals are shown in Fig. 12.

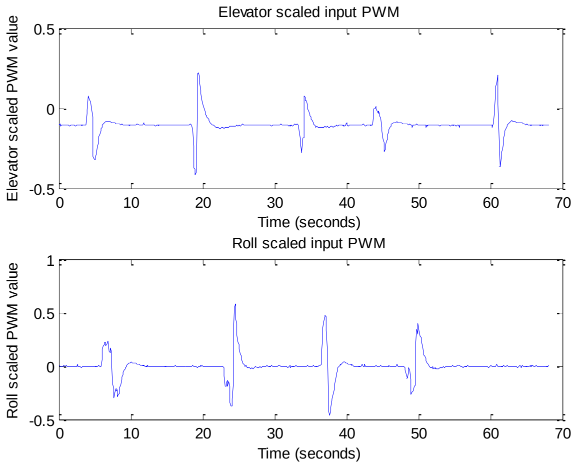

Figure 11: Input scaled PWM signals

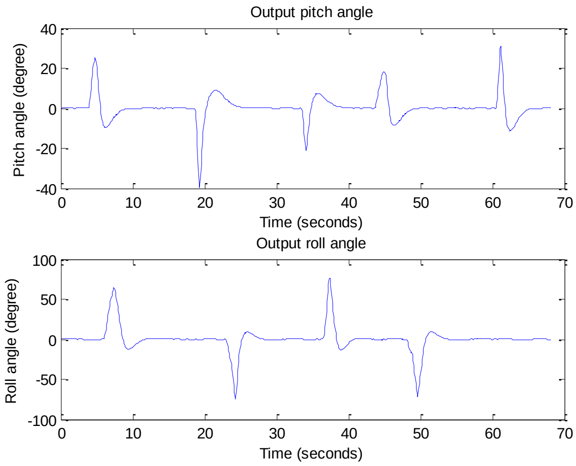

Figure 12: Output attitude signals

As seen in the Fig.12. , the simulated aircraft tends to reach zero pitch and roll. Zero level indicates the reference angle for system (Fig. 10). Positive and negative peaks in the output signals represent the moment that the RC remote controller sticks were released. When sticks are released, PID compensators are activated and the aircraft follows the reference angle which is zero degrees.

V. CONCLUSIONS AND FUTURE WORKS

In this paper we outlined the design of a pitch and roll stabilizer for a small fixed winged UAV using data obtained from Xplane simulations. These data are processed by a system identification process utilizing MATLAB and a dynamical model of the pitch and yaw behaviors are obtained. These models are used to construct PI controllers for these axes, and hardware-in-the-loop simulations using a custom two-degree-of-freedom moving platform confirm that the designed controllers successfully return the plane to a straight and level flight.

Simulated flight stabilizer can be used for a real flight because whole tested flight stabilize system is based on actual sensors which the moving hardware system platform includes. Our future goal is to test this stabilizer for a real flight. Another future goal is to design new autopilot algorithms in which a plane can follow predefined waypoints autonomously, and increasing autopilot system reliability and performance.

ACKNOWLEDGMENT

The authors thank TOBB ETU and TK3 Teknik for providing the resources used in this study, as well as the professional RC pilots Ali Can Çalış and Koray Kerimoğlu for their contributions to the flight tests.

REFERENCES

[1] HaiYang Chao, YongCan Cao and YangQuan Chen "Autopilots for Small Unmanned Aerial Vehicles: A Survey," International Journal of Control, Automation, and Systems (2010) 8(1):36-44 DOI 10.1007/s12555-010-0105-z

[2] J. M. Sullivan, ''Evolution or revolution? The rise of UAVs," IEEE Technology and Society Magazine, vol. 25, no. 3, pp. 43-49, 2006.

[3] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. Mclain, and M. Goodrich, ''Autonomous vehicle technologies for small fixed wing UAVs," AIAA J. Aerospace Computing, Information, and Communication, vol. 2, no. 1, pp. 92-108, 2005.

[4] System Identification Toolbox for Use With Matlab, The Math Works, Inc., User's guide, ver. 5, Nov. 2000, pp. 3–79.

[5] M. Liu, G. K. Egan, and Y. Ge, "Identification of attitude flight dynamics for an unconventional UAV," Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 3243–3248, 2006.

[6] YAO Junkui, QI Zaikang, LI Fugui, GUO Tao, " Study on UAV Flight Based Identification Technology," 29th Chinese Control Conference July 29-31, 2010, Beijing, China

[7] B. Kim, Y. Chang, and M. H. Lee, "System identification and 6-dof hovering controller design of unmanned model helicopter," JSME International Journal, vol. 49, pp. 1048–1057, 2006.

[8] H. Wu, D. Sun and Z. Zhou, "Model Identification of a Micro Air Vehicle in Loitering Flight based on Attitude Performance Evaluation". IEEE Trans. Robotics, vol. 20, pp. 702 712, August 2004.

[9] B. Mettler, T. Kanade, and M. B. Tischler, "System identification modeling of a small-scale unmanned rotorcraft for control design," J. Amer. Helicopter Soc., pp. 50–63, Jan. 2002.

[10] Santos, S.R.B.; Oliveira, N.M.F.; , "Test platform to pitch angle using hardware in loop," Frontiers in Education Conference, 2009. FIE '09. 39th IEEE , vol., no., pp.1-5, 18-21 Oct. 2009

[11] Ta-ming Shih; Ho-chung Chang; , "FPGA based hardware in the loop test platform of small size UAV," Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on , vol., no., pp.551-556, 15-18 Dec. 2009

Halim Korkmaz graduated from department of the electrical and electronics engineering in Hacettepe University, Ankara, Turkey in 2011.

He is working in TOBB ETU as teaching assistant, Ankara , Turkey since 2011. He is current M.Sc. student in TOBB ETU. He is studying on aerial control systems and developing various autopilot systems for unmanned aerial vehicles.

He is a member of IEEE.

Onur Baki Ertin graduated from department of the electrical and electronics engineering in TOBB ETU, Ankara, Turkey in 2011.

He is working in TOBB ETU as teaching assistant, Ankara , Turkey since 2011. He is current M.Sc. student in TOBB ETU. He is studying on aerial control systems and developing various autopilot systems for unmanned aerial vehicles.

Cosku Kasnakoglu obtained B.S. degrees from the Department of Electrical and Electronics Engineering and the Department of Computer Engineering at the Middle East Technical University (METU), Ankara, Turkey in 2000. He obtained his M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering at the Ohio State University (OSU), Columbus, Ohio, USA in 2003 and 2007. From 1996-2000 he worked as a researcher at the Turkish Scientific Council (TUBITAK), Information Technologies and Electronics Research Institute (BILTEN), from 2000-2007 he worked as a graduate researcher at OSU and in 2008 he joined TOBB University of Economics and Technology, where he is currently an associate professor in the Department of Electrical and Electronics Engineering. Dr. Kasnakoglu's current research interests include nonlinear control, flow control, unmanned air vehicles, dynamical modeling, adaptive control and linear parameter varying systems. He is a member of IEEE, AIAA, IACSIT and has published numerous scientific papers in respected journals and conferences.