Tail Motion Model Identification for Control Design of an Unmanned Helicopter

BERNARDO MORCEGO, ELEAZAR SALDÍVAR and FATIHA NEJJARI

Automatic Control Department, Universitat Politècnica de Catalunya,

TR11, Rambla Sant Nebridi 10, 08222 Terrassa, Spain.

e-mail: bernardo.morcego@upc.edu

Abstract

This paper explains the methodology developed to design the yaw control system (heading control system) of the α-SAC UAV. The problem of modeling and controlling the tail motion of this UAV along a desired trajectory is considered. First, the response data of the system are collected during special flight test and a linear time invariant model is extracted by identification techniques. Then, the control system is designed and implemented using a PID feedback/feedforward control method. The technique is tested in simulation and validated in the autonomous flight of the small scale helicopter.

Keywords: UAV, identification, yaw control.

1. INTRODUCTION

The Advanced Control Systems research group at Universitat Politècnica de Catalunya decided to develop a platform suitable for control algorithms test and design. The selected platform was a fleet of small scale unmanned autonomous helicopters (the acronym UAV is used for convenience). The long term research objective of the group is to design a fully autonomous and reliable helicopter autopilot, able to perform well with different UAV models and in different operating conditions.

Helicopters are mechanically complex and their dynamic behavior is very unstable, apart from being nonlinear and multivariable, with strong couplings between variables. Those are the main reasons why they require a lot of work by the pilot during flight and the main motivation for the design of a reliable autopilot. The viability of the small scale helicopter as a multipurpose research vehicle has driven great interest these days [1]-[6].

This paper explains the hardware and software system designed and mounted on the commercial R/C models the group has. This system converts the models into real UAVs. The paper also describes the control architecture developed and shows results of the methodology carried out to design the yaw control system (lateral control system). The small scale helicopter used for the results given here is a

Thunder Tiger Raptor 30 model, named hereafter α-SAC. Fig. 1 (left) shows a picture of the α-SAC carrying all the instrumentation during one of the experiments.

Figure 1: α-SAC helicopter in autonomous flight (left) and block diagram (right)

The paper is organized as follows: in section 2, the α-SAC hardware description is given; section 3 is devoted to the software design; section 4 describes the overall low-level autopilot control structure; the heading identification methodology is presented in section 5; the control design of the yaw dynamics and some results are given in section 6; finally, in Section 7 the main conclusions are summarized.

2. HIGH LEVEL SYSTEM DESCRIPTION

The UAV is configured around three subsystems: the helicopter, the command computer and the on-ground computer, as shown in fig. 1. The helicopter carries instrumentation, an on-board computer and a specialized microcontroller (µC). All the hardware devices are off the shelf commercial devices which were assembled, connected and programmed by the members of the research group.

The command computer is used to operate the helicopter manually. It is used as a safety device in case there is a system malfunction. The human pilot controls the helicopter with a radio control attached to a special interface in the command computer. The computer sends the pilot's commands to both the on-board computer and the µC via radio using a MaxStream XBeePRO modem.

The helicopter system receives the pilot's commands from the command computer with a 990.001 XBee module and distributes them to the on-board computer and the µC through serial/USB connections. Several sensors are also attached to the on-board computer: an Inertial Measurement Unit (IMU) that measures, among other variables, the orientation of the helicopter (pitch, roll and yaw); a barometer, used to obtain the altitude and a camera, that can be used to have an onboard perspective.

The µC is used to drive the helicopter's servos. The information transmitted to the on-board computer by the sensors and/or the command computer is processed and transmitted via a wired serial connection to the specialized microcontroller. Nevertheless, when the human pilot needs to take control of the system, he turns a safety switch of the radio controller on and then the µC ignores everything but the pilot's commands.

The on-ground computer connects to the on-board computer via standard Wi-Fi. The on-ground computer is used to monitor and control the UAV system.

3. SOFTWARE DESCRIPTION

The UAV system depends on the correct operation of several subsystems. Each of them is programmed for a specific task as described in the previous section. The most delicate part is the on-board computer since it is in charge of receiving data from sensors, receiving commands from the pilot (command computer), interacting with the supervising user, calculating control actions and finally issuing servo commands to the µC.

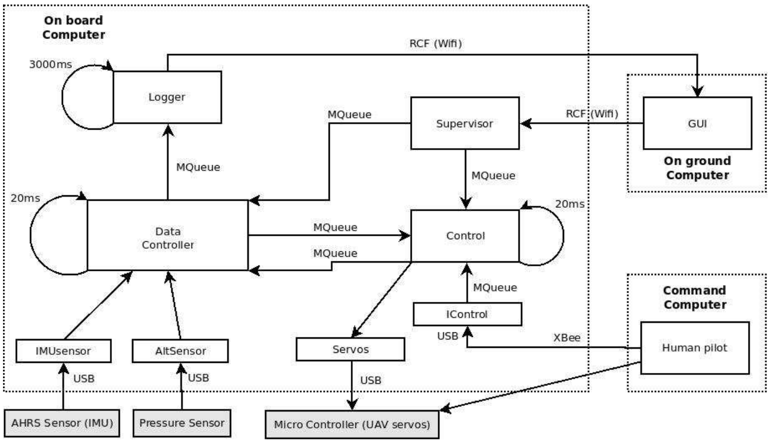

Figure 2: Diagram of the α-SAC software system. Dotted boxes represent applications, each of them running in a different computer; grayed boxes represent physical devices; and white boxes represent threads.

The on-board computer is a Linux machine with the real time kernel patch. Its processor is an Intel Atom Z530 at 1.6GHz with 1GB of board memory and the total weight of the computer is about 300gr.

The program running in the on-board computer is organized as a multithreaded process. The communication between threads is accomplished through message queues. This way it is possible to have different cycle times for each thread and different types of communication needs solved in a very robust and homogeneous way.

Sensor reading threads (labelled IMUsensor and AltSensor in fig. 2) are in charge of regularly collecting data from the inertial unit and the pressure sensor, respectively. They process the readings if necessary; for example, the altitude needs to be calculated from a filtered reading of the atmospheric pressure. Finally, they transmit the data through their message queues, which are of size one, allowing this way to send only the latest information from the sensors. Thread IControl behaves similarly but collecting data frames from the command computer instead of sensor readings.

The Logger thread buffers all the data the DataController thread sends. Its cycle time is the largest in the application and so is its priority (large numbers mean less priority). When it has collected enough data it sends the data through a standard Wi-Fi connection using the UDP protocol.

The Supervisor thread is in charge of controlling the application operation mode. It is an asynchronous thread since it only reacts to commands sent from the on-ground computer application. Typical actions that this thread carries out are: changing the control mode from automatic to manual, changing the controller from one type another one, setting new values for setpoints, etc.

Finally, the Control thread implements the control algorithms that drive one or more variables of the UAV autonomously. All the values of the helicopter servos are calculated each cycle, or obtained directly from the human pilot. Then they are sent to the Servos thread, which delivers them to the µC.

The software system introduces unavoidable delays to the control loop. When the on-board computer is controlling directly the servos of the helicopter, there is a maximum delay of 30ms between sensor data acquisition and servo movement. The sampling time is 20ms so the delay introduced by the control system is between one and two samples. When the human pilot is controlling the helicopter the delay can rise up to approximately 55ms, including all the communications and computing delays. This delay is short enough to allow an experienced pilot to control the helicopter.

4. CONTROL STRUCTURE

The control architecture of an autonomous helicopter is a controversial topic, according to the variety of options found in the literature. The reason for that is that helicopter control involves several variables (attitude, position, velocity) which can be dealt with in various ways.

Hierarchical control is often employed to separate attitude control from velocity or position control, as in [9]. In this case attitude control is in the lowest level loop.

Even attitude can be treated in various ways. In [10] and [11] an independent control loop for pitch, roll and yaw is proposed. On the opposite side [12] uses a MIMO controller with four control actions and 8 measures.

The control of a real helicopter is achieved through four flight controls operated by the pilot: the collective lever, the cyclic joystick, the lateral control pedals and the throttle. The first two are used to control the main rotor, the third one is for the tail rotor system and the last one is usually connected to a governor that depends on the amount of pitch in the main blades.

Although the effects of each control handle are not strictly related to only one variable in the system, it is true that there is a dominant relation between each control and the key variables of the system as follows: the cyclic stick works in two dimensions affecting roll and pitch angles; the collective lever is used to control the altitude of the craft; finally, the lateral control pedals change the yaw angle or heading of the helicopter.

The control structure used here for attitude control, shown in fig. 3, follows the structure of a real helicopter. There is a control loop for each actuator.

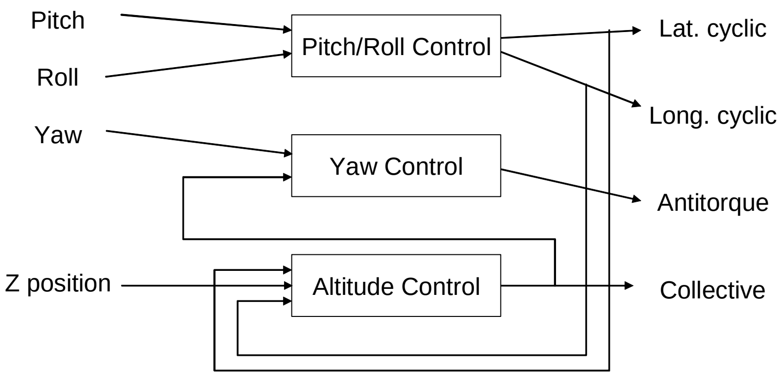

Figure 3: Control structure diagram. It shows the relations between the control variables that are taken into account.

5. IDENTIFICATION OF α-SAC HEADING DYNAMICS

To ensure the quality of data and obtain a model which represents the operation point dynamics reliably, flight experiments have to be carefully designed. The system is unstable and it cannot receive arbitrary input signals in open loop. Nevertheless, with the help of a skilled pilot, it is possible to place the helicopter in hover position and inject predesigned input sequences for a period of time during which the pilot does not operate on the commands. After that, or interrupting the experiment if the pilot feels it is necessary, the pilot recovers the control over the helicopter and it is safely returned to the ground. This way, all the experiments are virtually open loop experiments. Typical duration of an experiment is about 20s, which yields 1000 samples.

The data sets used as inputs are the following: a 10s MLBS (Maximum Length Binary Signal) spanning frequencies from 0.1 Hz to 25 Hz; a 5s MLBS spanning frequencies from 0.2 Hz to 25 Hz; a 20s multisine signal spanning frequencies from 0.5 Hz to 5 Hz; and a step signal. The amplitudes of the signals were tuned manually to make them as large as possible within the range of behaviors from which the pilot could recover the control of the helicopter. For the pedals input, u_2, binary and multisine signals were used. For the collective input, u_1, pulses inserted by hand were used. The amplitude was also tuned to be the largest. Fig. 4 shows sample fragments of the signals captured during the experiments.

System identification, based on the prediction error method, is applied to α-SAC UAV as a MISO system. The ident Matlab toolbox is used and the reader is addressed to [7] for a very comprehensive review of the methods used along this section. The auto-regressive moving average exogenous (ARMAX) models and the state space (SS) models are used to describe the dynamics of this system.

The SS and ARMAX models incorporate a polynomial that models additional information that is not a part of the model and is treated as noise. When the experiments are well designed and carried out, the data obtained has enough useful information of the models and this term and its effects are small.

The noise term of the ARMAX model obtained has a large influence in the output, which means that the resulting model does not describe well the helicopter.

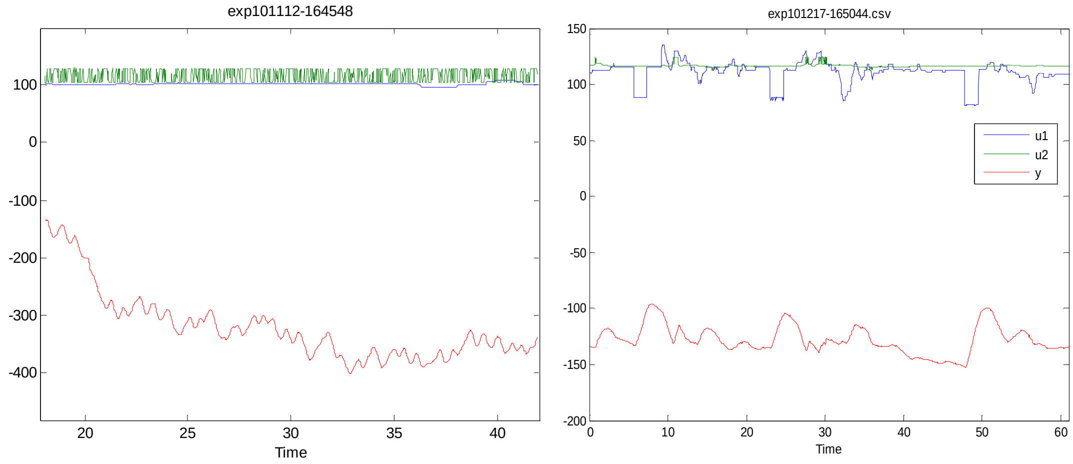

Figure 4: Fragments of the signals captured during identification experiments. Red line, *y*, is the yaw angle; green line, u_1, is the lateral control pedal signal; blue line, u_2, is the collective lever signal.

The selected model for the pedals-yaw relation is

$$\begin{aligned} &A(q)y(t) = B(q)u(t) + C(q)e(t) \\ &A(q) = 1 - 1.5(\pm 0.11)q^{-1} + 0.2(\pm 0.2)q^{-2} + 0.3(\pm 0.09)q^{-3} \\ &B_1(q) = -0.009(\pm 0.04)q^{-5} + 0.008(\pm 0.04)q^{-6} \\ &B_2(q) = 0.02(\pm 0.002)q^{-8} \\ &C(q) = 1 - 0.03(\pm 0.1)q^{-1} + 0.25(\pm 0.04)q^{-2} \end{aligned} \tag{1}$$

with a loss function of 0.643393 and an FPE of 0.647958.

For the collective-yaw relation we have:

$$\begin{aligned} &A(q)y(t) = B(q)u(t) + C(q)e(t) \\ &A(q) = 1 - 2.34(\pm 0.035)q^{-1} + 1.7(\pm 0.06)q^{-2} + 0.41(\pm 0.03)q^{-3} \\ &B_1(q) = -0.00009(\pm 0.0009)q^{-5} + 0.0009(\pm 0.0009)q^{-6} \\ &B_2(q) = 0.0006(\pm 0.0002)q^{-8} \\ &C(q) = 1 - 1.3(\pm 0.02)q^{-1} + 0.66(\pm 0.02)q^{-2} \end{aligned} \tag{2}$$

with a loss function of 0.0248903 and an FPE of 0.0250202.

The other method of identification was done considering a state space pre-fixed structure based on the physics model of the helicopter's tail dynamics and on [8]. The identification algorithm used was PEM. The advantage of using this type of identification is that additional information of the system known beforehand can introduce in the model.

The initial structure is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & Ts \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (3)$$

where x_1 represents the angle, x_2 the angular velocity, u_1 the collective input, u_2 the pedals input and y the yaw angle. The seed parameters are taken from [9]. $a_{21}=-.3$, $a_{22}=.9$ $b_{21}=.1$ and $b_{22}=.2$.

In general these values show that the collective (u_1) has a smaller influence in the yaw output than the pedals (u_2) and that the system has and integrator effect. This linear model is assumed to be good enough when the output and inputs are small, the helicopter is hovering and the angular velocities are small.

From the experiments of the pedals-yaw it is obtained:

$$A = \begin{bmatrix} 1 & 0.02 \\ -0.00376 & 0.88796 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0.04716 & 1.009 \end{bmatrix} \quad (4)$$

with a loss function of 0.639837 and an FPE of 0.642947.

For the collective-yaw relation it is obtained:

$$A = \begin{bmatrix} 1 & 0.02 \\ -0.021203 & 0.89778 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ -0.082775 & 0.56623 \end{bmatrix} \quad (5)$$

with a loss function 0.0254125 and FPE 0.0255121.

The two model structures are evaluated in table I. The parameters compared are the 5-step prediction fit, FPE, loss function, the noise-output/u-output coefficient and the model order; the first parameter should be as high as possible, and for the rest of the parameter the lower the number is, the better.

Table 1: ARMAX and SS Model Comparison

	ARMAX		SS	
	Collective	Pedals	Collective	Pedals
Fit id-data	95.57%	97.18%	96.02%	97.25%
Fit val-data	96.39%	91.43%	95.97%	92.32%
Noise-Y/u-Y	53	26.8	37	20.3
FPE	0.02502	0.6479	0.02551	0.6429
Loss	0.02489	0.6433	0.02541	0.6398
Model Order	3	3	2	2

Based on this comparison, that shows the slightly better results of the SS model compared to the ARMAX one, the SS model will be used along the rest of this work.

Fig. 5 shows the time responses of the 5-step predicted output of the chosen model and the measured output.

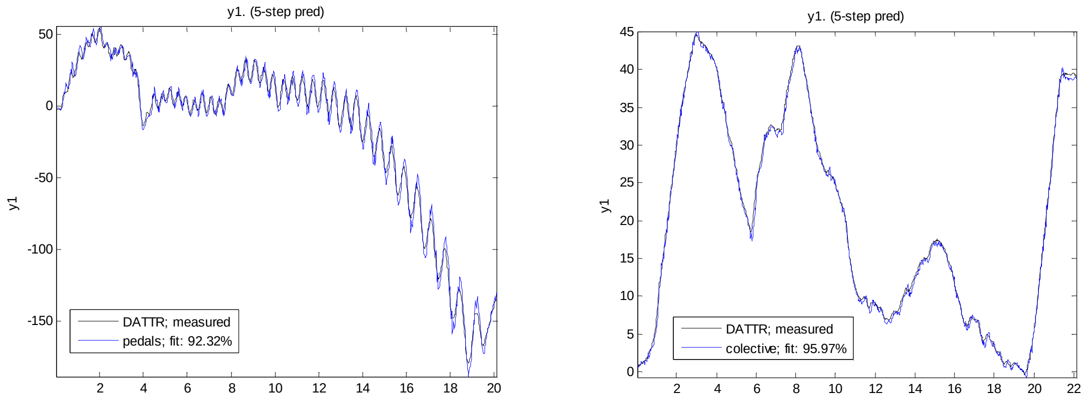

Figure 5: Measured and model 5-step ahead predicted output for the pedals-yaw relation (left) and for the collective-yaw relation (right). Validation data set is used.

The two transfer functions obtained from the SS model are:

$$G_1(z) = \frac{-0.001656}{z^2 - 1.898z + 0.8982}, \quad G_2(z) = \frac{0.02018}{z^2 - 1.888z + 0.888} \tag{6}$$

6. CONTROL DESIGN

A PID feedback controller is used for the pedals-yaw relation, and a feedforward controller for the collective-yaw relation. The control diagram is shown in fig. 6. Disturbance rejection is managed by the feedback loop but it also helps correcting the disturbances caused by changes on the cyclic commands. Disturbances correspond mainly to air gusts. The changes in collective are compensated for with the feedforward controller.

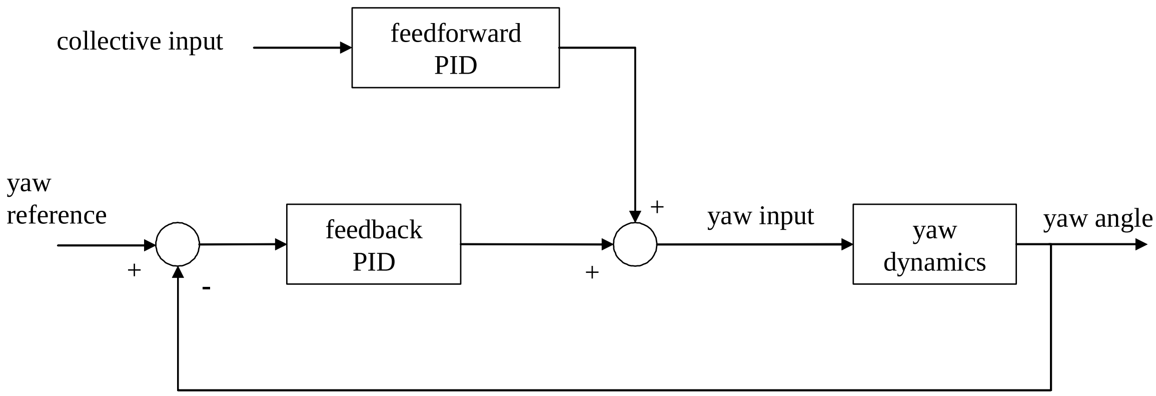

Figure 6: Block diagram of the yaw controller, considering two inputs, one output, a feedback controller and a feedforward controller.

The controller is developed according to the following specifications: rise time of 1 sec., settling time smaller than 5 sec. and overshoot between 5% and 8%. The design is made focusing on disturbance rejection and the actuator's ranges. The controller is designed using the sisotool of Matlab and implemented and tested in simulation with Simulink.

The amplitudes for the collective and pedals step signals are taken from the usual maximum variations that the inputs have in flight. The wind gust step value (load perturbation) is deduced with the help of the pilot. A step with amplitude of 5 produces 45 degrees per second change and is considered a high strength wind. A breeze corresponds to a value of 1 and causes an angular change of 9 degrees per second. The amplitude selected for the simulation is 2.5.

The feedforward controller is

$$FF = \frac{0.001656\, z^2 - 0.003126\, z + 0.00147}{0.02018\, z^2 - 0.0383\, z + 0.01813} \quad (7)$$

This part was first modeled as a pure static gain (FF_{gain}=0.0147). It showed a very poor performance and its effect was barely noticeable, so the complete transfer function of the feedforward controller is used.

The feedback controller transfer function is:

$$PID_{F350} = \frac{3.2z^2 - 5.87z + 2.69}{z^2 - z + 0.000912} \quad (8)$$

Its rise time is 0.201s, the overshoot is 61.3%, the settling time is 2.85s, the gain margin is 9.5 dB and the phase margin is 26.1˚.

Fig. 7 shows the response of the system under control. The overshoot is higher than expected because this simulation includes delay terms not accounted for in the design.

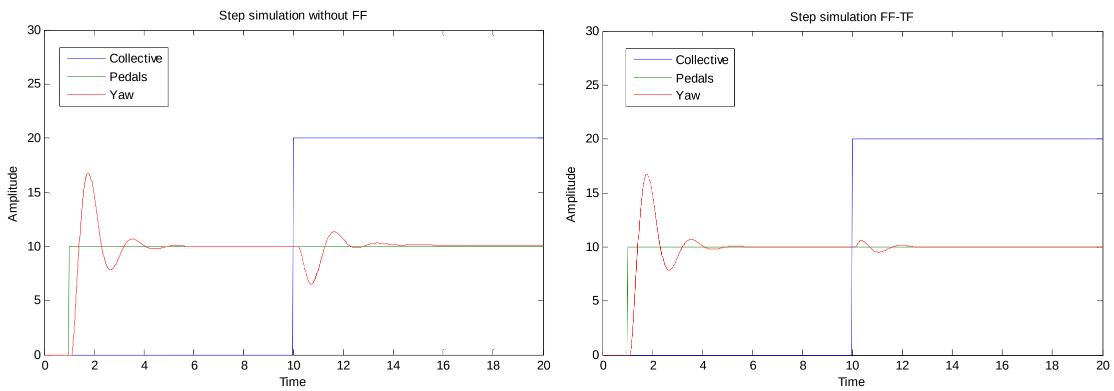

Figure 7: Closed loop response. At time 1s there is a yaw set point change and at time 10s there is a collective change. Left figure corresponds to a feedback controller and right figure corresponds to a feedforward-feedback controller.

Figure 8 shows the feedforward and feedback controllers working in the helicopter in flight. The controller is activated at time 20. After it is working we can see that the yaw angle stays around the reference of -165°; from this point until time 105 the “pedals” input is only modified by the controllers output. If we disregard the peaks that take place at time 60, that correspond to a bounce with the ground, the angle does not exceed 5 degrees of error. If we take close attention to the two input signals, we can see that the pedals’ input reacts to the movements of the collective; it is more noticeable at time 30, 55 (before the bump) and 75. Although Wind and air gusts were present during the experiment, they could not be measured, but their effects are reflected in the tracking of the reference. A simplified, but similar effect was seen in the simulation of the wind gust.

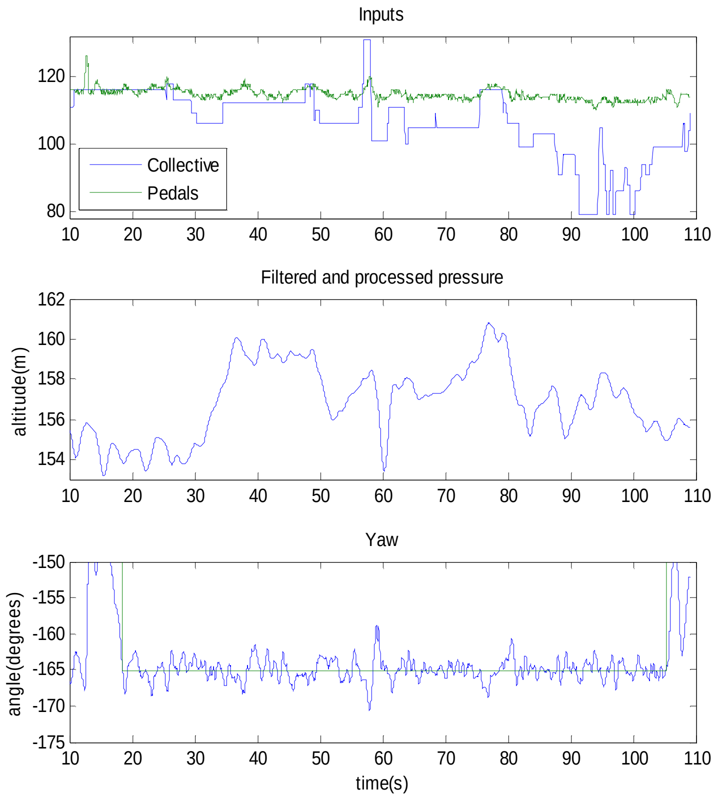

Figure 8: Experimental results of yaw controller. Top figure shows the inputs of the system; middle figure shows the altitude of the helicopter; and bottom figure shows the yaw angle and the set point.

7. CONCLUSIONS

This paper resents the control design of the yaw of an unmanned helicopter system using multi-loop control approach. System identification experiments are applied to model the yaw motion of the UAV then ARMAX and SS models are identified and validated. Accuracy of the model is verified by comparing real flight and simulation data obtained during the experiments.

The controller for the yaw system consists of a feedforward and a feedback controller. Essentially the feedforward section consists in a mathematic cancellation of the signal that affects the system. In the simulations this controller shows good attenuation of the perturbation but not a complete cancellation due to the delays of the signals.

The feedback controller is obtained by tuning the PID controller parameters taking into account the specifications proposed. The control design is validated experimentally showing good performance.

ACKNOWLEDGMENT

This work was supported in part by the Spanish Exploration and Innovation Ministry under project DPI2008-00403 (TEOMAPIN) and DPI2008-01996 (HYFA). The authors also want to thank Carlos Sanlley for his dedication and skilled piloting; without him most of the identification experiments would not had been able to be carried out.

REFERENCES

[1] G. D. Padfield, *Helicopter Flight Dynamics: The theory and Application of Flying Qualities and Simulation Modeling*, Second Edition. AIAA Education Series, Oxford Blackwell Publishing, 2007

[2] R. Austin, *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*. John Wiley & Sons, 2010

[3] X. Wang, X. Zhao and M. Tan, *Modeling, Identification and Robust Control of Yaw Dynamics of Small-Scale Unmanned Helicopters*, The Key Laboratory of Complex Systems and Intelligence Science Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

[4] D. McFarlane and K. Glover, “A Loop Shaping Design Procedure Using H∞ Synthesis,” *IEEE Transactions on Automatic Control*, vol. 37, Jun. 1992, pp. 759-769, doi:10.1109/9.256330.

[5] E. N. Sanchez, H. M. Becerra and C. M. Velez, “Combining fuzzy, PID and regulation control for an autonomous mini-helicopter,” *Information Sciences, v*ol. 177, issue 10, pp 1999-2022, May 2007

[6] B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft*, Kluwer Academic Publishers, Boston, MA, USA, 2003.

[7] L. Ljung, T. Glad, *Modeling of Dynamic Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 2004

[8] D. H. Shim, H. J. Kim and S. Sastry, “Control System Design for Rotorcraft-based Unmanned Aerial Vehicles using Time-domain System Identification,” *Proceedings of the 2000 IEEE International Conference on Control Applications*, Anchorage, 2000

[9] J. del Cerro, “Arquitectura Abierta para el Control Autónomo y Teleoperado de un Mini-Helicòpter”, PhD Dissertation, Universidad Politécnica de Madrid, Spain, 2001.

[10] C. P. Sanders, P.A. DeBitetto, E. Feron, H. F. Vuong and N. Leveson, "Hierarchical control of small autonomous helicopters," *Proceedings of the 37th IEEE Conference on Decision and Control*, vol.4, pp.3629-3634, 1998.

[11] Z. Fang, J. Wu, P. Li, "Control system design and flight testing for a miniature unmanned helicopter," *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on* , vol., no., pp.2315-2319, 2008.

[12] G. Cai, B. M. Chen, X. Dong, T. H. Lee, “Design and implementation of a robust and nonlinear flight control system for an unmanned helicopter”, *Mechatronics*, In Press, 2011.