ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

CONTROL OF YAW ANGLE IN A MINIATURE HELICOPTER

Diego F. Sendoya-Losada and Jesús D. Quintero-Polanco
Department of Electronic Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia
E-Mail: diego.sendoya@usco.edu.co

ABSTRACT

In this paper the modeling, identification and control of the yaw movement in a miniature coaxial helicopter is presented. A comparison between a PID design based on the CACSD tool (FRTool) and an auto-tuning algorithm (KCR) is performed, for both set point trajectory and disturbance rejection. An additional analysis based on root-locus techniques is done in order to verify the limits in the specified closed loop performance. Both controllers were successful and the system remained stable throughout the experiments.

Keywords: auto-tuning, CACSD, coaxial helicopter, UAV.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAV) have gained interest from the academic community worldwide, especially due to their numerous applications in civil and military applications. Civil applications include among others scientific, emergency and surveillance missions, as well as industrial applications (Meyer *et al*., 2009).

A special class of UAV is represented by very small vehicles that can be easily carried by a human. This class bears the name of micro aerial vehicles (MAV). There are various flying platforms, e.g. fixed-wing, quad-rotors, helicopters, coaxial helicopters, flying blimps each with their own pros and cons. Miniature coaxial helicopters and quad-rotors represent the best alternatives for building a miniature UAV platform for indoor flight (Bouabdallah *et al*., 2007). The applications of these autonomous flying platforms include surveillance in indoor areas (e.g. commercial centers), search and rescue inside a building in case of a disaster, fire detection, etc. Helicopters pose special abilities: hovering, vertical takeoff-landing, low speed cruise, pirouette; however, this increased maneuverability makes them more difficult to control and maintain stability. In general, the helicopter dynamics can be characterized by a MIMO (Multi Input Multi Output) strongly coupled system with non-linearity and intrinsic instability.

This paper is focused on the design of a controller for the yaw movement of a miniature coaxial helicopter using a new auto-tuner algorithm. Auto-tuners are algorithms that enable the user to rapidly obtain a reasonable set of parameters for a PID controller, without prior identification of the system. There are several methods for automatic tuning, which can be divided in two categories: i) based on identification of one point of the process frequency response and ii) based on the knowledge of some characteristic parameters of the open-loop process step response. The identification of a point in the frequency response of the process can be performed either using a proportional regulator, which brings the closed-loop system to the stability boundary, or by a relay, forcing the process output to oscillate.

The validation of the proposed auto-tuner was performed through experiments on the UAV platform, built using Cots (Commercial off-the-shelf) components. The architecture of the UAV platform was presented in detail in (Neamtu *et al*., 2010).

The structure of this paper is as follows: in the next section a brief overview of the UAV platform is presented, then the new auto-tuner principle and algorithm is summarized, the last section deals with the results obtained using the proposed approach and the conclusions and future works.

2. MATERIALS AND METHODS

2.1 MINIATURE COAXIAL HELICOPTER

The helicopter used for building the UAV system is a commercially available coaxial helicopter, depicted in Figure-1. The use of a coaxial helicopter as a miniature UAV for indoor use is quite popular and has been reported in several papers (Bouabdallah *et al.,* 2007, Chen *et al*., 2007, Miller *et al.,* 2008 and Schafroth *et al*., 2009).

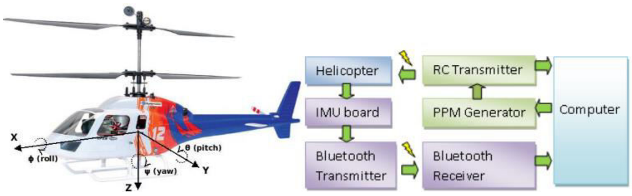

Figure-1. Coaxial helicopter and the block scheme of the experimental setup.

Table-1. Specifications of the helicopter.

Main rotor diameter	460mm
Weight	410g
Length	510mm
Width	110mm
Height	260mm
Motors	code 370 (2 installed)
Battery	11.1V 800mAh Li-polymer

In Figure-1, it can be seen a schematic of the components that constitute the UAV. The arrows in the figure represent the direction of the flow of data between components. Continuous lines represent wired connections and dotted lines represent wireless communication. The components of the UAV are as follows: 1) an aircraft to be

ARPN Journal of Engineering and Applied Sciences

controlled - the BigLama coaxial helicopter (see Figure-1 and Table-1); 2) an on-board IMU (Inertial Measurement Unit) with sensors to determine the attitude of the aircraft-gyroscopes, accelerometers, pressure sensors; 3) bidirectional wireless communication - a modified RC transmitter for sending the commands to the helicopter and Bluetooth communication for sending the data measured by the sensors to the computer; and 4) a ground-based computer.

The control algorithm runs on the ground-based computer. This is a multi-threaded program which accomplishes the following tasks: timing of the control algorithm; read the commands sent by the user using the joysticks and the keyboard; read, process and store data coming from the sensors; compute the command to the helicopter based on the current settings and the information coming from the sensors. In the current implementation, only the yaw movement of the helicopter can be automatically controlled; i.e. the other movements (pitch, roll, altitude) are manually controlled by the operator using the joysticks on the RC transmitter. The process to be controlled, i.e. the movement of the coaxial helicopter, is a MIMO coupled system. This system can be approximated using a SISO (Single-Input Single-Output) by ignoring the couplings in the system.

First input to the process is the aileron - controls the angle of the helicopter around the X axis (see Figure-1) which is known as roll angle. Second, input is the elevator - controls the angle of the helicopter around the Y-axis, which is called pitch angle. The third input is the throttle - controls the power to the two motors that spin two counter-rotating rotors.

The primary effect of the throttle is the change in altitude of the helicopter. The fourth input is called rudder - controls the speed of rotation of the helicopter around the Z-axis. This paper is focused on controlling the yaw movement, thus by sending commands using the rudder input and using as feedback the measurements of a gyroscope that measures the angular speed around the Z-axis. A coaxial helicopter is equipped with two counter-rotating rotors. In order to prevent the helicopter from spinning, the two rotational speeds of the two counter rotating rotors must be equal so that the generated torques has equal magnitude but opposite directions, hence they cancel each other. When yaw movement is required, one of the rotors should spin slower and the other rotor will spin faster so that they exert the same lift force.

In the following, an electro-mechanical model for the yaw movement of a coaxial helicopter is proposed. The functioning of a DC motor is described by:

$$u = u_e + Ri + L\frac{di}{dt}$$

where i is the current through the circuit, u is the voltage at the input of the motor, u_e is the voltage induced by the movement of the rotor in the electric field of the permanent magnet, R is the electrical resistance and L is the inductance of the coil. If the inertia of the blades of the helicopter is ignored, since the dynamics of the helicopter body are much slower than the dynamics of the rotor blades, the torque produced by the one rotor is:

$$\tau = B\omega$$

Where$\tau$the torque is generated by the DC motor on its shaft, ω is the angular rate of the rotor and B is the viscous friction coefficient. The equations that link the mechanical part to the electrical part are:

$$U_e = K_e\omega$$

$$\tau = K_t i$$

WhereK_e is a constant that links the induced voltage with the angular rate of the rotor and K_t is a constant that links the current flowing through the coils of the motor to the generated mechanical torque?' Combining these equations and applying the Laplace transform, it can be obtained:

$$\tau(s) = \frac{\frac{K_t}{L}}{s + \frac{R}{L} + \frac{K_e K_t}{BL}} u(s)$$

The two counter rotating rotors exert two opposite direction torques on the helicopters body. Here, Γ denotes the torque exerted on the helicopters body, which is the result of the difference$\Gamma = \tau_{top} - \tau_{bot}$, where τ_{top} and τ_{bot} are the torques produced by the upper rotor and the lower rotor respectively.

The relation between the torque exerted on the helicopters body and the difference of the input voltages Δu to the two DC motors is:

$$\Gamma(s) = \frac{2\frac{K_t}{L}}{s + \frac{R}{L} + \frac{K_e K_t}{BL}} \Delta u(s)$$

From the equilibrium condition, the torque exerted on the helicopter's body should be equal to:

$$\Gamma = J_h\frac{dr}{dt} + B_h r$$

Where r the angular rate of the helicopter's body is, J_h is the moment of inertia of the body and B_h is the friction coefficient of the body with the air. If it is considered that the helicopter rotates at relatively slow angular rates, then the air friction becomes negligible and the second term in equation can be dropped. Having this in mind, the relation between the angular speed of the helicopter r and the difference in input voltage Δu is obtained:

$$\Gamma(s) = \frac{2\frac{K_t}{L}}{s + \frac{R}{L} + \frac{K_e K_t}{BL}} \frac{1}{J_h s} \Delta u(s)$$

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Miniature helicopters come equipped with a gyroscope for controlling the yaw movement. The measurements of the gyroscope are introduced in a proportional controller, which can be manually tuned. In this case, Kdenotes the gain of the controller and the final transfer function for yaw movement becomes:

$$H_{yaw} = \frac{r(s)}{r_{desired}(s)} = \frac{\frac{2K_t}{LJ_h}}{s^2 + \left(\frac{R}{L} + \frac{K_e K_t}{BL}\right)s + \frac{2KK_t}{LJ_h}}$$

Where$r(s)$ is the measured angular speed of the helicopter around the Z-axis and $r_{desired}(s)$ is the angular speed that is sent to the helicopter using the rudder input (this value is scaled for values between -1 and 1).

In order to determine the real model of the yaw movement, a "black box" identification was used. For this, the software package called CIFER (Comprehensive Identification from Frequency Response) was used (Tischler *et al.*, 2006). This software package is suitable for miniature helicopter identification as reported in (Adripawita *et al.*, 2007, Ivler *et al.*, 2008, Mettler *et al.*, 2000 and Mettler *et al.*, 1999).

In order to perform the identification experiment, the helicopter was brought into hovering mode by manually controlling the helicopter. After the helicopter was stabilized, a sine-sweep signal with the frequency varying from 1 Hz to 5 Hz and amplitude between ±0.2 was sent to the helicopter on the rudder input, as in Figure-2 (left).

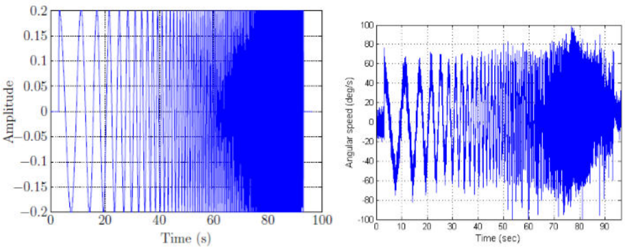

Figure-2. Chirp experiment: input (left) and output (right) signals.

The obtained output signal (see Figure-2 right) was preprocessed. First, the signal is re-sampled and later low-pass filtered to remove the noise produced by the motors and the helicopter's vibrations. Next, using CIFER, the transfer function for the yaw movement can be obtained:

$$H_{yaw} = \frac{172130}{s^2 + 19.15s + 712.3} e^{-0.0288\,s}$$

This transfer function is in accordance to the model derived based on electro-mechanical principles because it has 2 poles and no zero. The time delay that appears in this transfer function is due to Bluetooth communication. Other two procedures were employed for identification: i) using a PRBS (Pseudo-Random Binary Signal) as input and the identification toolbox from Matlab® for processing; and ii) using a new method called Chirp-TFA (Ionescu *et al.*, 2010) for obtaining the frequency response.

2.2 AUTO-TUNING ALGORITHM

The present auto-tuning algorithm referred to as the KCR auto-tuner (De Keyser *et al.*, 2010), is based on the prior art where two relay-based PID auto-tuners have been presented: the Kaiser-Chiara auto-tuner and the Kaiser-Rajka auto-tuner. Notice that the development of the PID controller using auto-tuning techniques does not require a-priori knowledge of the system.

Briefly, the KCR algorithm is based on the approximation of a closed loop response by a dominant second order transfer function $T(s)$ with unitary gain given by:

$$T(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n + {\omega_n}^2}$$

With ω_n the natural frequency and ζ the damping factor. From this equation the relationship between the closed loop percent overshoot ($OS\%$) and the peak magnitude M_p in frequency domain can be obtained (Nise, 2007):

$$\%OS = 100e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$

$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$$

By specifying the allowed overshoot in the closed loop, it follows that the closed loop transfer function must fulfill the condition:

$$|T(j\omega)| = \left|\frac{G(j\omega)}{1 + G(j\omega)}\right| = M_p$$

With$G(j\omega)$ the open loop transfer function of both the process and the controller. Re-writing this equation in its complex form with R the real part and I the imaginary part:

$$|T(j\omega)| = \frac{R(\omega) + jI(\omega)}{[1 + R(\omega)] + jI(\omega)}$$

By taking $|T(j\omega)|^2$ results that:

$$(R + c)^2 + I^2 = r^2$$

Where$c = \frac{{M_p}^2}{{M_p}^2-1}$ and$r = \frac{M_p}{{M_p}^2-1}$,which is nothing else than the equation of a (Hall-) circle with radius r and center in $\{-c, 0\}$ (Nise, 2007). In order to have a peak magnitude, only those circles with M >1 are of interest. Intersection with the unit circle is achieved by using this last equation and the condition: $R^2 + I^2 = 1$, hence solving for R and I yields:

$$R = 0.5\frac{1 - 2{M_p}^2}{{M_p}^2}$$

$$I = -\frac{\sqrt{{M_p}^2 - 0.25}}{{M_p}^2 - 0.5}$$

The phase margin is given by $\tan(PM) = \frac{|I|}{|R|}$ thus:

$$PM = tan^{-1}\frac{\sqrt{{M_p}^2 - 0.25}}{{M_p}^2 - 0.5}$$

Specifying PM does not suffice to guarantee a good closed loop performance in all situations. Therefore, the next step is to determine the cross-over frequencyω_c; i.e. the frequency where the loop frequency response crosses the 0 dB line.

$$|R(j\omega_c).G(j\omega_c)| = 0\ dB = 1$$

If the settling time of the closed loop is specified, then using $T_s = \frac{4}{\zeta\omega_n}$ the bandwidth frequencycan be obtained:

$$\omega_{BW} = \frac{4}{\zeta T_s}\sqrt{(1-\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}$$

From a previous work, $\omega_{BW} \approx 1.5\omega_c$ and the generalization to higher order systems gives $\omega_c \le \omega_{BW} \le 2\omega_c$. By having the cross-over frequency ω_c, a sinusoid with period $T_c = \frac{2\pi}{\omega_c}$ can be applied to the process to obtain the output:

$$G(j\omega_c) = Me^{j\varphi} = M(\cos\varphi + j\sin\varphi)$$

Using the transfer function analyzer algorithm (Ionescu *et al.*, 2010). The task is now to find the controller parameters such that the specification for settling time T_s and overshoot percent $\%OS$ is fulfilled. The PID controller is written as:

$$R(s) = K_p\left[1 + \frac{1}{T_i s} + T_d s\right]$$

which for the cross-over frequency becomes:

$$R(j\omega_c) = K_p\left[1 + j\left(T_d\frac{2\pi}{T_c} - \frac{1}{T_i\frac{2\pi}{T_c}}\right)\right]$$

Starting from the controller frequency response in this equation, the loop frequency response is given by:

$$\begin{aligned}R(j\omega_c).G(j\omega_c) &= 1.e^{j(180°+PM)}\\ &= \cos(-180° + PM)\\ &\quad + j.\sin(-180° + PM) = -a - jb\end{aligned}$$

with$a = \cos PM$ and $b = \sin PM$, schematically shown in Figure-3.

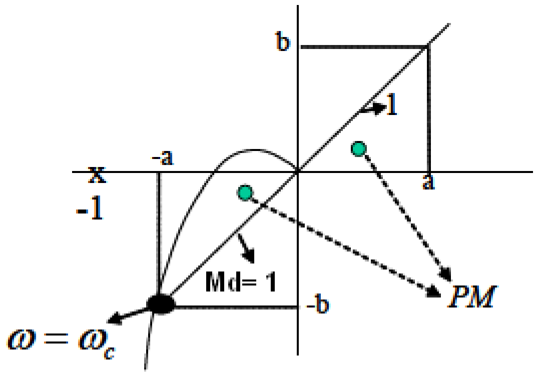

Figure-3. Schematic of the loop frequency response.

Thus, the controller is given by:

$$R(j\omega_c) = K_p\left[1 + j\left(T_d\omega_c - \frac{1}{T_i\omega_c}\right)\right] = K_p(1 + j\alpha)$$

$$\begin{aligned}K_pM[(\cos\varphi - \alpha\sin\varphi) + j(\sin\varphi + \alpha\cos\varphi)]\\ = -(\cos PM + j\sin PM)\end{aligned}$$

From the real and imaginary parts:

$$\alpha = \frac{\tan PM - \tan\varphi}{1 + \tan PM\tan\varphi} = \tan(PM - \varphi) = T_d\omega_c - \frac{1}{T_i\omega_c}$$

$T_i = 4T_d$is chosen to produce real zeros in the controller; however, other choices can be made. Consequently, last equation becomes:

$$T_d\omega_c - \frac{1}{4T_d\omega_c} = \tan(PM - \varphi)$$

From where

$$T_i = T_c\frac{\sin(PM - \varphi) \pm 1}{\pi\cos(PM - \varphi)}$$

which gives only one positive result.

$$(K_pM)^2.(1 + \alpha^2) = 1$$

With $1 + \alpha^2 = 1 + \tan^2(PM - \varphi) = \frac{1}{\cos^2(PM-\varphi)}$, which gives the K_p controller parameter:

ARPN Journal of Engineering and Applied Sciences

$$K_p = \pm \frac{\cos(PM - \varphi)}{M}$$

with only one positive result.

It is always possible to find one of the three special cases of a PID controller: P, PI or PD. Using the KCR algorithm.

In order to design a PD controller from, it is necessary to cancel the integral action. This can be done by putting $T_i = \infty$, instead of $T_i = 4T_d$.

$$T_d = T_c \left[\frac{\tan(PM - \varphi)}{2\pi}\right]$$

Similarly, the PI controller is found when the derivative action is equal to zero.Thus, with$T_d = 0$ yields:

$$T_i = -T_c \left[\frac{1}{2\pi \tan(PM - \varphi)}\right]$$

It is also possible to design a P controller, if both the derivative action and the integral action are cancelled. This occurs if $T_i = \infty$and $T_d = 0$.

$$\tan(PM - \varphi) = 0$$

which gives that $PM = \varphi$. And, it follows that:

$$K_p = \frac{1}{M}$$

It is important to notice that the P auto-tuner can be calculated just based on the magnitude value of the sinusoidal signal at ω_c frequency. However, due to the fact that the parameter $PM = \varphi$, the $\%OS$ cannot be specified as an independent design parameter, leaving only the settling time T_s as a specification parameter.

3. RESULTS AND DISCUSSIONS

The adopted control strategy is a cascade controller as depicted in Figure-4.

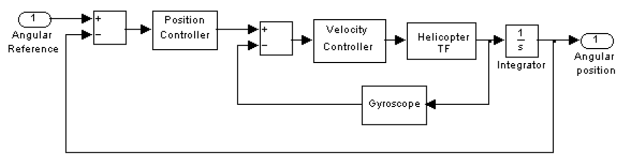

Figure-4. Closed loop control scheme.

Using computer aided control system design tools, such as the FRtool (De Keyser *et al.*, 2006), a PID controller for angular velocity controlwas designed. The specifications for the controller were: robustness 50%, settling time 0.6 seconds and maximum 10% overshoot. The resulting controller parameters were:$K_p = 9.11x10^{-4}$, $T_i = 0.032$, and $T_d = 0.008$.

By integrating the angular speed using a software integrator, the yaw angle of the helicopter is obtained. The target consists in designing a position control loop for the yaw angle, while the velocity control remains with fix parameters as presented above. The closed loop transfer function obtained from the inner velocity loop, which will be used to design the position controller is:

$$H_{vel} = \frac{0.04s^2 + 5.012s + 156.6}{0.032s^3 + 0.653s^2 + 27.81s\ 156.6}$$

Two different controllers are compared based on simulated and real data.

3.1 P CONTROLLER

Firstly, the design based on the knowledge of the transfer function model is developed using the FRtool package. The yaw angle controller meets the following specifications: robustness 50%, settling time of 1 second and maximum 20% overshoot. Hence, a proportional controller was used with the gain:

$$K_p = 6.52$$

Applying the same design specifications for the KCR auto-tuner, it follows that the requirement for the cross-over frequency is $\omega_c = 3.1416\ \text{rad/s}$. A sinusoidal signal with ω_c frequency was applied to the system, from where the magnitude (M) and phase (φ) were determined as 0.1319, and 44.43°, respectively. Since the P controller depends solely on the magnitude M, then:

$$K_p = 7.58$$

The performance comparison between these two P controllers is depicted in figure 5, showing that both are similar. It is therefore remarkable that the PID KCR auto-tuner leads to similar results as the model-based-design PID controller.

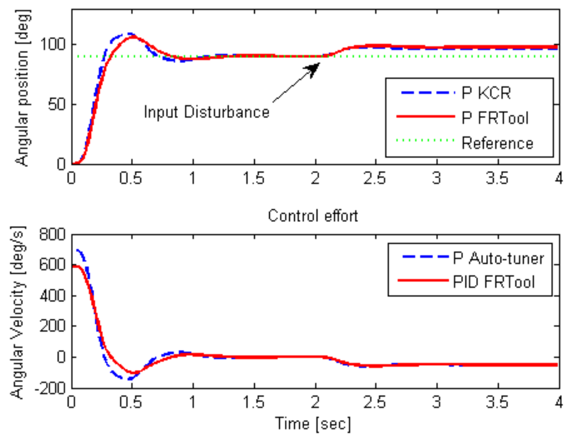

Figure-5. Simulation results using P controllers.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

3.2 PID CONTROLLER

The next step was to design a PID controller, in order to remove the steady-state error and to obtain a smooth and fast response with the minimal control effort. The best performance using the FR tool model-based design was reached with the following parameters: $K_p = 2.895$, $T_i = 0.6911$, and $T_d = 0.1727$.

The cross-over frequency parameter ω_c to tune the KCR auto-tuner based on the desired settling time, it follows that the requirement for the cross-over frequency is $\omega_c = 2.0944\ \mathrm{rad/s}$. A sinusoidal signal with ω_c frequency was applied to the system, from where the magnitude (M) and phase (φ) were determined as 0.4645, and 17.44º, respectively. The auto-tuner formulas then results in: $K_p = 3.108$, $T_i = 0.924$, and $T_d = 0.231$.

The comparison of the performance of both PID controllers is given in Figure-6. Similar conclusion as for the P controllers can be drawn; the auto-tuner performs similarly to the model-based design PID.

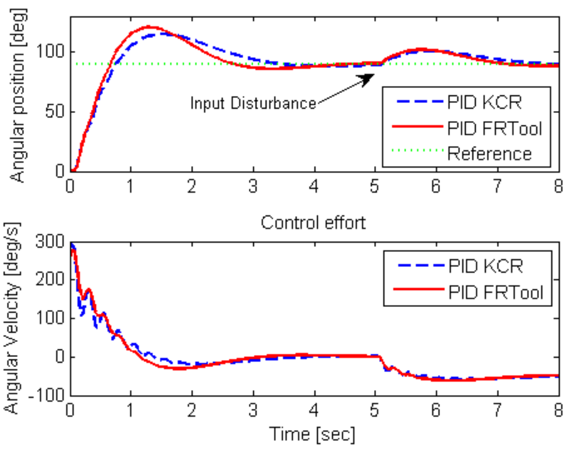

Figure-6. Simulation results using PID controllers.

It is possible to conclude that both controllers achieve the setpoint, while preserving the stability of the system. Nevertheless, in order to make clear why it was not possible to decrease the settling time for the PID controllers, a root-locus analysis was done.

The root-locus plot from Figure-7 enables observations upon the differences between both controllers, where the black line represents the position where the dominant poles in closed loop must be located in order to reach a settling time of 1 second.

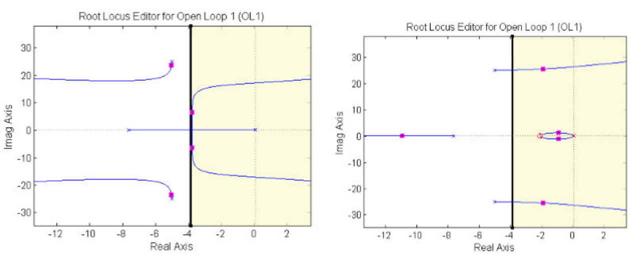

Figure-7. Root-locus diagram for the P controller (left) and for the PID controller (right).

The root-locus for the PID controller shows the limitation in the settling time, due to the location of the poles and zeros. The dominant poles in closed loop for the P controller are on the left side of the limit line, i.e. for this controller the settling time is less than 4 seconds. The small oscillations in the transient for the PID controllers are due to the high values of the imaginary part of the closed loop poles.

Also, some overshoot can be expected, introduced by the presence of the zero. It is also important to mention, the fact that despite the P controller has a faster response, it cannot reject the input disturbances, due to a lack of integrator action on the forward path (see Figure-5). In other hand, the disturbance rejection is not a problem for the KCR-PID controller (Figure-6).

3.3 REAL PLANT EXPERIMENTS

Finally, the KCR auto-tuners were tested on the miniature helicopter UAV. The performance of the auto-tuners in the real setup is similar to the one presented by simulation. This shows the reliability of the algorithm and the fact that the KCR achieves the desired setpoint, as depicted in Figure-8.

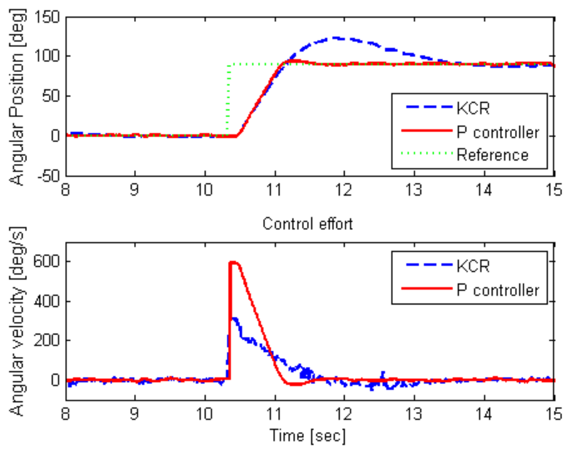

Figure-8. KCR auto-tuner response in the real system.

An increased delay can be observed in the experimental data, but this does not affect the stability of the controller. The origin of this increased (variable) delay is the communication, via the RF transmitter and the Bluetooth connection, which may receive interference from other wireless applications.

3.4 CONCLUSIONS

Two computer-aided-design controllers and two auto-tuned controllers for yaw movement were successfully applied on a miniature coaxial helicopter. The analysis of the results shows that the performance of the auto-tuner is equally good to that of a model-based PID control design. Notice that the auto-tuner does not require any a-priori system identification. Both P and PID controllers could follow a setpoint trajectory, but only the PID was able to reject input disturbances. Based on the root-locus representation, it was shown that in the current

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

setup, a PID control for the angular position cannot be faster than 3 seconds for the settling time. Next step in this study is to investigate whether measuring acceleration can provide a better closed loop control and faster settling times, while maintaining the stability of the system. Also, efforts to obtain auto-tuners for pitch and roll axis are currently in progress, allowing a full three degrees of freedom closed loop control.

REFERENCES

Meyer J., De Plessis F. and Clarke W. 2009. Design considerations for long endurance unmanned aerial vehicles. vol. Aerial Vehicles, ISBN: 978-953-7619-41-1, pp. 443-497, InTech, Croatia.

Bouabdallah S., Becker M. and Siegwart R. 2007. Autonomous minituare flying robots: coming soon. IEEE Robotics & Automation Magazine. Vol. pp. 88-98.

Neamtu D., Deac R., De Keyser R., Ionescu C. and Nascu I. 2010. Identification and Control of a Miniature Rotorcraft Unmanned Aerial Vehicle (UAV). in Proc IEEE intconf on Automation Quality and Testing Robotics (AQTR 2010). pp. 1-6.

Chen L. and McKerrow P. 2007. Modelling the Lama Coaxial Helicopter. in Proc of the Austral-asianConf on Robotics and Automation. Brisbane, pp. 1-9, Univ of Wollongong.

Miller S. M., MacCurdy R. B., Kidd W. R. and Hudson J. M. 2008. Stabilization and Control of a Micro-scale Helicopter. Cornell University, Ithaca, NY, 14850.

Schafroth, D., Bermes, C., Bouabdallah, S., and Siegwart R. 2009. Modelling and system identification of the muffly micro helicopter. in J of Intel and Robotic Syst, proc 2nd Int. Symp on UAVs, Reno, U.S.A. pp.27- 47.

Tischler M. B. and Remple R.K. 2006. Aircraft and rotorcraft system identification. IEEE Control Systems Magazine, American Institute of Aeronautics and Astronautics. pp. 101-103.

Adripawita W., Ahmad A.S. and Semibiring J. 2007. Automated flight test and system identification for rotary wing small aerial platform using frequency response analysis. in International Conference on Intelligent Unmanned Systems. pp. 50-56.

Ivler C. M. and Tischler M.B. 2008. System Identification modeling for flight control design. in RAeS Rotorcraft Handling-Qualities Conference, Univ of Liverpool.

Mettler B., Tischler M. and Kanade T. 2000. System Identification of model-scale helicopter. tech. report CMU-RI-TR-00-03, Robotics Institute, Carnegie Mellon University.

Mettler B., Tischler M. and Kanade T. 1999. System Identification of small-size Unmanned Helicopter Dynamics. At the American Helicopter Society 55th Forum, Montreal, Quebec, Canada.

Ionescu C., Robayo F., De Keyser R. and Naumovic M. 2010. The frequency Response Analysis Revisited", in Proc. IEEE 18th MeditConfon Control and Automation, Marrakesh, Marocco. pp. 1441- 1446.

De Keyser R. and Ionescu C. 2006. FRTool: a Frequency Response Tool for CASCD in Matlab. Proc IEEE Conf on Comp Aided Control Syst Design. pp. 2275-2280.

Nise N.S. 2007. Control Systems engineering. 4th ed, Wiley.