Tuning of controller for an aircraft flight control system based on particle swarm optimization

Emre Kiyak
Department of Avionics, Anadolu University, Eskisehir, Turkey

Abstract

Purpose – This study aims to present a method for the conceptual design and simulation of an aircraft flight control system.

Design/methodology/approach – The design methodology is based on particle swarm optimization (PSO). PSO can be used to improve the performance of conventional controllers. The aim of the present study is threefold. First, it attempts to detect and isolate faults in an aircraft model. Second, it is to design a proportional (P) controller, a proportional derivative (PD) controller, a proportional-integral (PI) controller and a fuzzy controller for an aircraft model. Third, it is to design a PD controller for an aircraft using a PSO algorithm.

Findings – Conventional controllers, an intelligent controller and a PD controller-based PSO were investigated for flight control. It was seen that the P controller, the PI controller and the PD controller-based PSO caused overshoot. These overshoots were 18.5, 87.7 and 2.6 per cent, respectively. Overshoot was not seen using the PD controller or fuzzy controller. Steady state errors were almost zero for all controllers. The PD controller had the best settling time. The fuzzy controller was second best. The PD controller-based PSO was the third best, but the result was close to the others.

Originality/value – This study shows the implementation of the present algorithm for a specified space mission and also for study regarding variation of performance parameters. This study shows fault detection and isolation procedures and also controller gain choice for a flight control system. A comparison between conventional controllers and PD-based PSO controllers is presented. In this study, sensor fault detection and isolation are carried out, and, also, root *locus*, time domain analysis and Routh–Hurwitz methods are used to find the conventional controller gains which differ from other studies. A fuzzy controller is created by the trial and error method. Integral of squared time multiplied by squared error is used as a performance function type in PSO.

Keywords Optimization, PID control, Flight control, Fault detection/diagnosis, Fuzzy control/logic

Paper type Research paper

Nomenclature

Definitions, acronyms and abbreviations

P controller = Proportional controller
PD controller = Proportional + derivative controller
PI controller = Derivative + integral controller
PSO = Particle swarm optimization.

Introduction

An automatic control system consists of a controller, an actuator, a sensor and a plant. In a closed loop control system, the difference between input and feedback is fed to the controller so as to reduce error and to bring the output of the system to a desired value. The actuator is a power device that produces input to the plant according to the control signal, so that the output will approach the reference input. The sensor converts the output variable into another suitable variable. An advantage of this system is the fact that the use of feedback makes the system response relatively insensitive to external disturbance and internal variation.

The aim of the controller of closed loop control systems is to produce an output following input. Conventional controllers are widely used for this aim, with certain variations according to the system structure. These controllers have advantages and disadvantages. A proportional (P) type controller's main advantage is its simplicity. The advantage of the integral (I) controller is that the output is proportional to the accumulated error. Thus, error can be eliminated using it. The advantage of the derivative (D) controller is that the controller will provide large connections before the error becomes large. A P-type controller's main disadvantage is that there may be a fixed steady state error. The disadvantage of the I controller is that the system is less stable if the pole is added at the origin. The disadvantage of the D controller is that if the error is constant it will not produce a control output (Nelson, 1998).

Min *et al.* suggest a control system design method for an autonomous helicopter, using a hybrid PSO algorithm. Their proposed hybrid PSO algorithm combines the basic PSO algorithm and a sequential quadratic programming algorithm to improve convergence speed (Min *et al.*, 2006). Meng *et al.* (2010) propose a new reconfigurable flight control system

The current issue and full text archive of this journal is available on Emerald Insight at: **www.emeraldinsight.com/1748-8842.htm**

Aircraft Engineering and Aerospace Technology: An International Journal
88/6 (2016) 799–809

[ISSN 1748-8842]
[DOI 10.1108/AEAT-02-2015-0037]

Received 14 February 2015
Revised 10 June 2015
Accepted 12 June 2015

Aircraft Engineering and Aerospace Technology: An International Journal
Volume 88 · Number 6 · 2016 · 799–809

design methodology based on an eigenstructure assignment with constrained output feedback and a proposed PSO algorithm. Wenyue *et al.* (2011) use a multiple optimization algorithm, a cultural algorithm based on particle swarm optimization (PSO) to apply flight control law clearance. Their algorithm provides a more accurate performance in function optimization compared with a traditional PSO. They claim their proposed algorithm provides an effective approach in the search for worst combinations of parametric model uncertainties during the process of flight clearance (Wenyue *et al.*, 2011). Khan *et al.* (2011) propose an optimized reconfigurable control design methodology by separating the control command distribution task from the flight controller for different types of fault handling. They claim their proposed strategy improves flight control performance in normal and abnormal cases. A PSO is used to produce virtual command signals (Khan *et al.*, 2011). Khan *et al.* (2011) present a genetic algorithm-based modular reconfigurable control strategy for an over-actuated nonlinear aircraft system. Their control law is based on a multi-input, multi-output (MIMO) linear quadratic regulator strategy to produce virtual command signals. They use a natural evolution-based optimization technique in modular control design (Khan *et al.*, 2012). Jingping *et al.* (2011) apply a face searching allocation algorithm to design a reconfigurable control system for an aircraft model. Duan *et al.* (2013) present a predator–prey PSO algorithm for identifying parameters of a unmanned combat aerial vehicle (UCAV) flight control system to reduce the workload of designers during the process of designing a complicated UCAV control system. In this approach, a new fitness function, proposed during the design procedure, is proven appropriate and efficient by a series of comparative experimental results (Duan *et al.*, 2013). Montazeri-Gh *et al.* (2013) presented an application of PSO to achieve tuning of an integrated flight and propulsion control. The gains of the controllers are tuned by PSO, where the tuning process is formulated as an optimization problem (Montazeri-Gh *et al.*, 2013). El-Saady *et al.*(2014) present an aircraft automatic landing controller using PSO to improve the performance of an automatic landing system and to guide an aircraft to a safe landing. Abaspour *et al.* (2013) propose introducing an optimal fuzzy logic control law for a non-linear control method. They use PSO to optimize the membership functions' parameters of their proposed design (Abaspour *et al.*, 2013). Alagoz *et al.* (2013) present a stochastic, multi-parameter, divergence optimization method for auto-tuning of proportional-integral-derivative (PID) controllers according to a fractional-order reference model. Their study is to approximate the step response of a real closed-loop flight control system to the response for a smoother and more precise experience (Alagoz *et al.*, 2013). Romero *et al.* (2011) propose a new auto-tuning algorithm for proportional-integral (PI) and PID controllers based on relay experiments to minimize load disturbance integral error by maximizing integral gain, subject to a desired phase margin and a minimum required gain margin constraint (Romero *et al.*, 2011). Ömürlü and Yildiz represent stiffness control by means of an independent joint fuzzy-proportional derivative (PD) control algorithm with gain scheduling to be used as a fly-by-wire flight control unit. Their model and real system responses are compared, using stiffness control, so that the model is valid for control design trials. Responses are compared with alternative control algorithms such as fuzzy-PD, self-tuning fuzzy PD and PD controllers (Ömürlü and Yildiz, 2011). Oner *et al.* (2012) present a mathematical model and vertical flight control algorithms for a new tilt-wing unmanned aerial vehicle. The vehicle is capable of vertical take-off and landing. The mathematical model of the vehicle is obtained using Newton–Euler formulation. A gravity-compensated PID controller is designed for altitude control, and three PID controllers are designed for attitude stabilization of the vehicle. The performance of these controllers is found to be quite satisfactory as demonstrated by indoor and outdoor flight experiments (Oner *et al.*, 2012). Demirci and Kerestecioglu designed a controller method for linear MIMO systems. A sliding-mode controller is reconfigured in case of system faults. Faults are detected with a residual vector generated from a standard linear observer. Once a fault has been detected, a fault distribution matrix can be obtained and used to update the corrective or equivalent control parts of the sliding mode controller. As a result, fault tolerant adaptive controllers keep the system performance within acceptable limits or at least avoid the system winding up (Demirci and Kerestecioglu, 2005). Karasakal *et al.* (2005) developed a self-tuning method for fuzzy PID controllers. In their tuning method, an input scaling factor corresponding to the derivative coefficient and an output scaling factor corresponding to the integral coefficient of the fuzzy PID controller are adjusted using a fuzzy inference mechanism with a new input called normalized acceleration. The results of the implementation have been compared with those of a classic fuzzy PID controller without a tuning mechanism. Hajiyev and Caliskan propose an approach to detect and isolate aircraft sensor and control surface/actuator failures occurring in an aircraft control system. An extended Kalman filter (EKF) has been developed for nonlinear flight dynamic estimation of an F-16 fighter. Failures in sensors and control surfaces/actuators affect the characteristics of the innovation sequence of the EKF. Theoretical results are confirmed by simulations carried out on a nonlinear dynamic model of the F-16 aircraft (Hajiyev and Caliskan, 2005). Caliskan and Hajiyev (2013) investigated techniques on aircraft icing identification based on a neural network, batch least-squares algorithm, Kalman filtering and H_∞ parameter identification and made comparisons.

In this study, the roll angle control of a flight control system model is developed using PSO. When the flight control model is used, sensor faults continue to be a major hurdle for flight control system health management to reach its full potential. Because of this, model-based fault detection and isolation techniques can be used to generate a fault indicating signal called a residual. As a result, sensor faults are detected and isolated. One of the most important tasks in the design of the control system is to determine the structure of the controller and the elements. Traditional PID controllers and fuzzy controllers are widely used to reduce or eliminate steady-state error and to improve the dynamic response of the system. However, manual tuning of these controllers is time consuming, tedious and generally leads to poor performance. Instead of these controllers, a PSO algorithm for tuning the optimal PD controller can be used. This approach has superior features, including easy implementation, stable convergence characteristics and good computational efficiency over conventional methods.

In this study, the detection and isolation of faults on an aircraft model are carried out. Second, utilizing certain controllers, a flight control system is used for the comparison of the P controller, PD controller, PI controller, fuzzy controller, and, finally, the PD controller-based PSO is investigated by utilizing the roll control system of an aircraft. The system is considered to be composed of a comparator, controller and aircraft equation of motion. The sensor is considered to be a perfect device and is represented as a unity feedback.

Fault detection and isolation for sensor failures

Steady state representation is useful for analysis:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

Here, A is the system matrix, B is the control distribution matrix, C is the measurement distribution matrix, x is the state vector and u is the input vector. An observer can be used to generate an estimate of the state based on measurements of the system output and the system input. If the system has a fault, an observer, based on model-based fault detection and isolation technique, can be used. The structure of the observer is described as (Ammar, 2000; Stevens and Lewis, 2003; Blanke *et al.*, 2006; Solak, 2001):

$$\dot{z}(t) = Fz(t) + Gy(t) + Lu(t) \tag{2}$$

$F \in R^{nxn}$ is the observer dynamic matrix, $G \in R^{nxn}$ is the measurement distribution matrix, $L \in R^{nxm}$ is the control distribution matrix and $z(t) \in R^{nx1}$ is the observation vector. If both sides of the first of the equation (1) are pre-multiplied by a matrix T, equation (3) is obtained:

$$\begin{aligned} \dot{z}(t) - T\dot{x}(t) = {} & Fz(t) + Lu(t) + GCx(t) - TAx(t) \\ & - TBu(t) \end{aligned} \tag{3}$$

Equation (3) can be reorganized as equation (4):

$$\begin{aligned} \dot{z}(t) - T\dot{x}(t) = {} & F(z(t) - Tx(t)) + (FT - TA + GC)x(t) \\ & + (L - TB)u(t) \end{aligned} \tag{4}$$

Assuming that the matrix can be constructed such that the following equations are satisfied:

$$FT - TA + GC = 0 \tag{5}$$

$$L - TB = 0 \tag{6}$$

Equation (4) can be obtained as:

$$e(t) = e^{Ft}e(0) \tag{7}$$

If the matrix F is selected, such that all F's eigenvalues are in the left half of the complex plane, the solution goes to zero asymptotically as equation (8):

$$\lim_{t\to\infty} e(t) = 0 \tag{8}$$

So, it follows in the steady state:

$$\lim_{t\to\infty} z(t) = \lim_{t\to\infty} Tx(t) \tag{9}$$

Conventional controllers and particle swarm optimization

The most important task in the design of the control system is to determine the structure of the controller and the elements. For this purpose, PID-type controllers are widely used. The PID controller is used to reduce or eliminate steady-state error. Additionally, it improves dynamic response. The derivative effect of the controller adds a finite zero to the open-loop plant transfer function and improves transient response. The integral effect of the controller adds a pole at the origin and reduces steady-state error. The PID controller transfer function is given by:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{10}$$

If the model is not known exactly, control procedure is not produced easily. Fuzzy logic controllers can be used for this purpose. Here, membership functions and a rule table are used. The input variables in a fuzzy control system are mapped into membership functions, known as fuzzy sets. The process of converting a crisp input value to a fuzzy value is called fuzzification. A microcontroller or computer makes decisions regarding what action to take based on rules. The results of all the rules are defuzzified to find a crisp value using one of several methods.

PSO was introduced by Kennedy and Eberhart (1995). This algorithm is based on population search. The original PSO algorithm is described as:

$$v_{id} = v_{id} + c_1 rand()(p_{id} - x_{id}) + c_2 Rand()(p_{gd} - x_{id}) \tag{11}$$

$$x_{id} = x_{id} + v_{id} \tag{12}$$

Here, c_1 and c_2 are positive constants, *rand*() and *Rand*() a re two random functions in the range [0,1]; $X_i = (x_{i1}, \cdots, x_{iD})$ shows the i. particle; $P_i = (p_{i1}, \cdots, p_{iD})$ shows the best position of the i. particle; g is the index of the best particle and $V_i = (v_{i1}, \cdots, v_{iD})$ shows the rate of the velocity (the position change) for particle i. Equation (11) consists of three parts. These are the momentum part, cognitive part and social part, respectively. The momentum part says that the velocity cannot be changed abruptly and is changed from current velocity. The cognitive part says velocity is changed by its own flying experience with the social part saying that velocity is changed by group flying experience. Equation (12) shows the position update of the particles. Shi and Eberhart add a new parameter into the original PSO algorithm (Shi and Eberhart, 1998).

This algorithm is described as:

$$v_{id} = wv_{id} + c_1 rand()(p_{id} - x_{id}) + c_2 Rand()(p_{gd} - x_{id}) \tag{13}$$

Here w is the inertia weight. It balances between global and local search abilities and eliminates the requirement of carefully setting maximum velocity V_{max}. Equation (12) is used the same in this developed algorithm.

Few studies relating to the use of a PSO algorithm flight control system are realized. Bassi *et al.* (2011) present an artificial intelligence method of a PSO algorithm for tuning the optimal PID controller parameters for industrial processes. The tuning method of Ziegler–Nichols was applied in PID tuning and results were compared with PSO-based PID for optimum control. They claim simulation results show that the PSO-based optimized PID controller was capable of providing an improved closed-loop performance over the Ziegler–Nichols tuned PID controller parameters (Bassi *et al.*, 2011). Meng *et al.* (2010) apply flight control law clearance using a multiple optimization algorithm–the cultural algorithm–based on particle swarm optimization. Their algorithm provides a more accurate performance in function optimization compared to traditional PSO. They claim the results indicate that the proposed algorithm provides an effective approach in the search for worst combinations of parametric model uncertainties during the process of flight clearance (Meng *et al.*, 2011). Hassan *et al.* (2013) use PSO to automatically tune membership

Figure 1 Roll angle of the flight control system block diagram

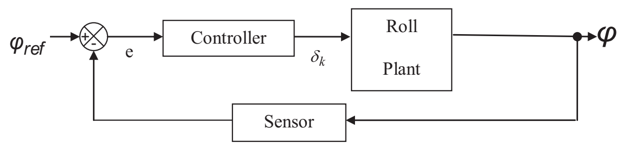


Figure 2 Roll rate response of aircraft

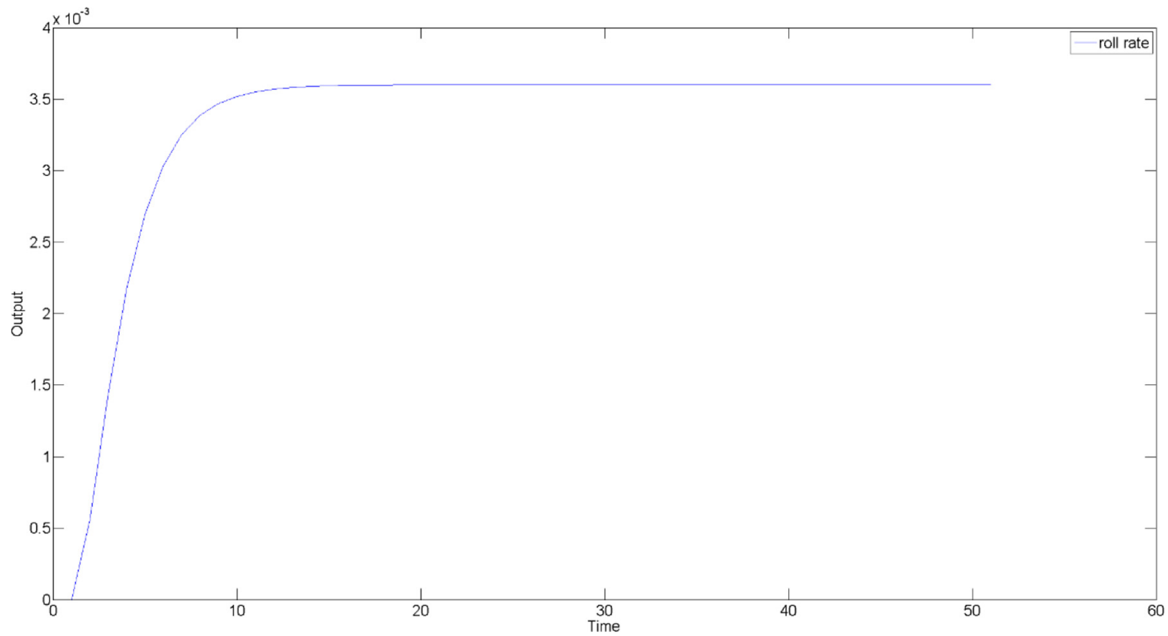


Figure 3 States of aircraft

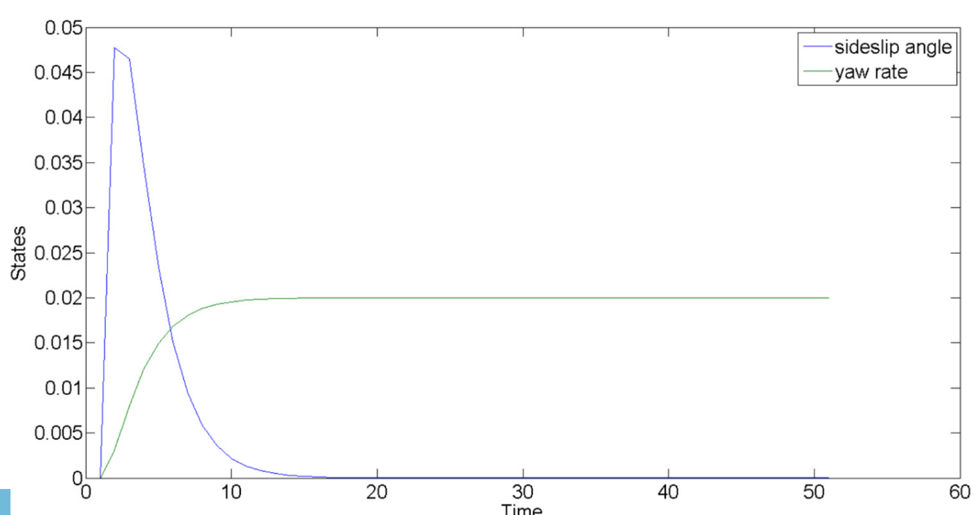


Figure 4 Estimated states

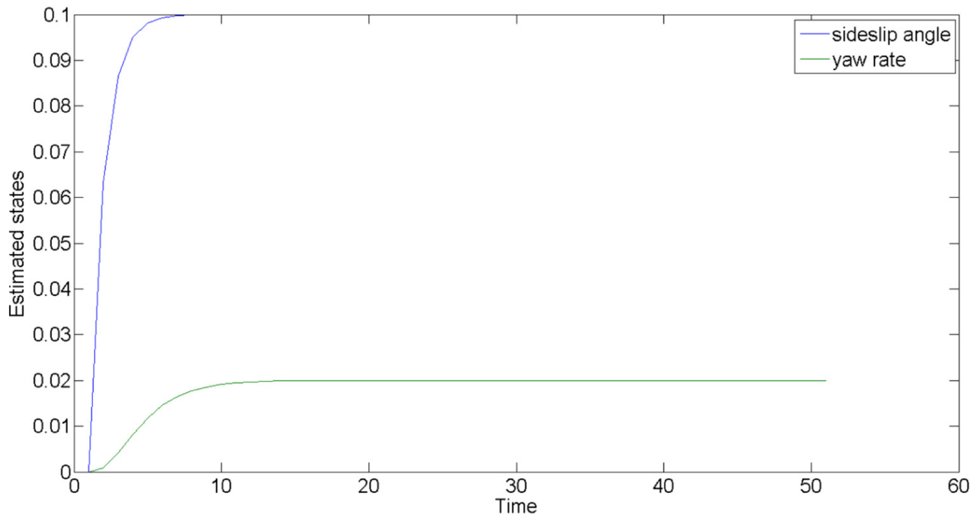


Figure 5 Residuals

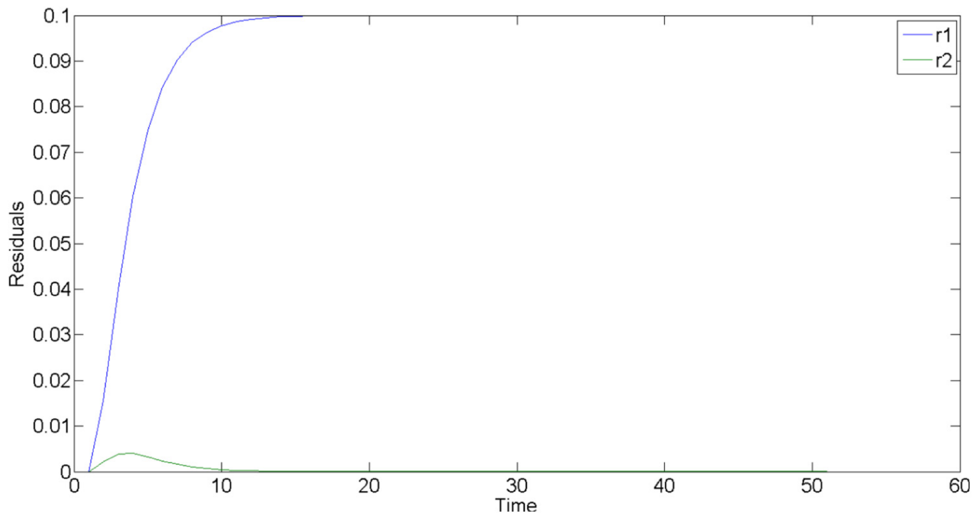
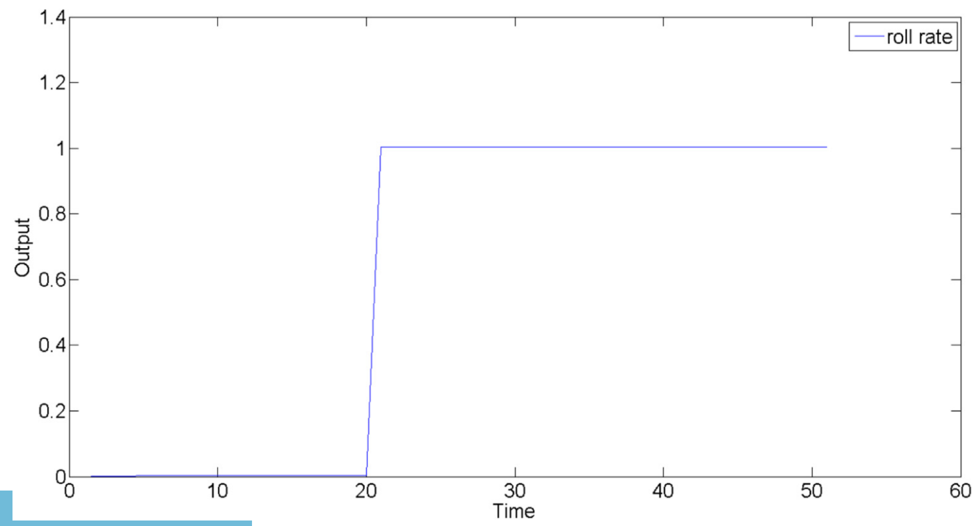


Figure 6 Output when a sensor fault occurs



functions of flight control guidance law. Optimal fuzzy logic guidance law is compared with proportional navigation guidance law and classical fuzzy logic guidance law. They claim the simulation results show that optimal fuzzy logic guidance law performs better than other guidance laws and that the introduced design performs well in the existence of noisy measurements (Hassan *et al.*, 2013).

In this study, sensor fault detection and isolation are carried out differently from the abovementioned studies. Also, the root *locus* method, time domain analysis and Routh–Hurwitz

Figure 7 Fault detection and isolation

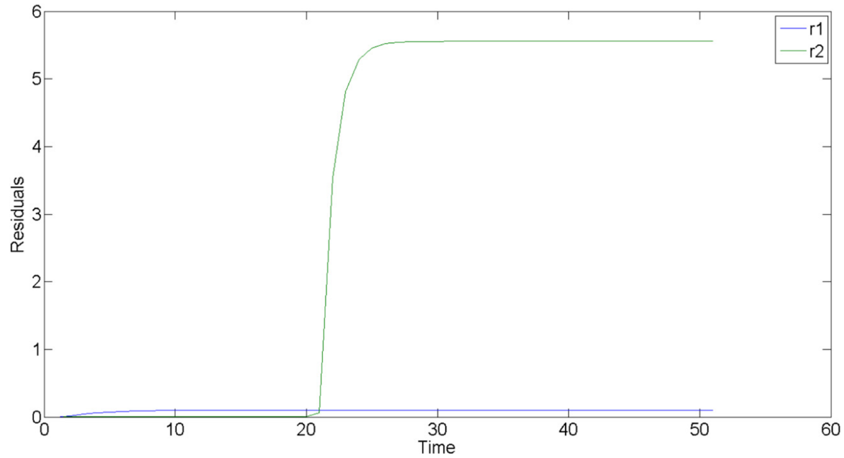


Figure 8 The unit step response for $K_p = 1.257$

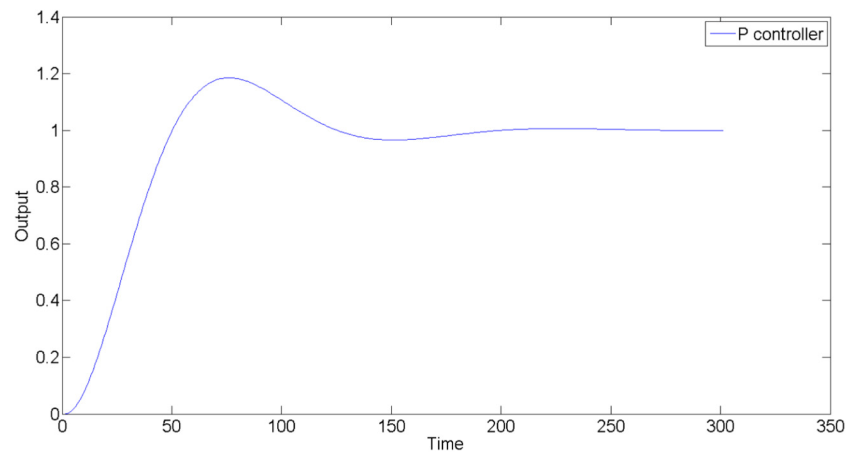
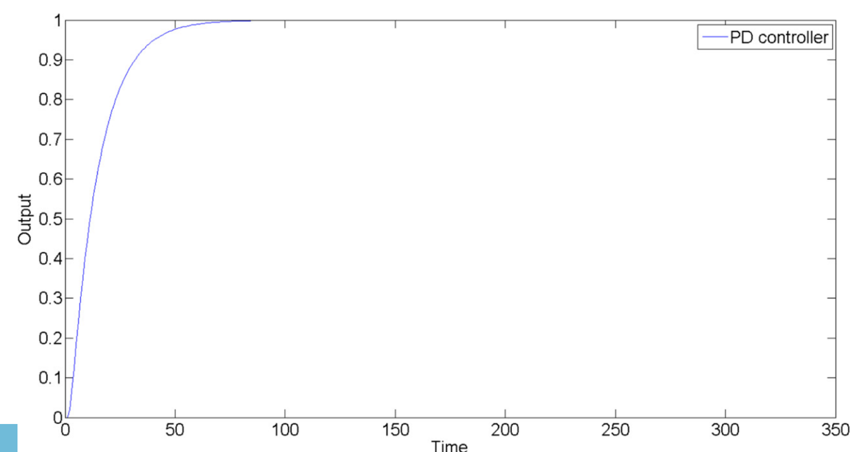
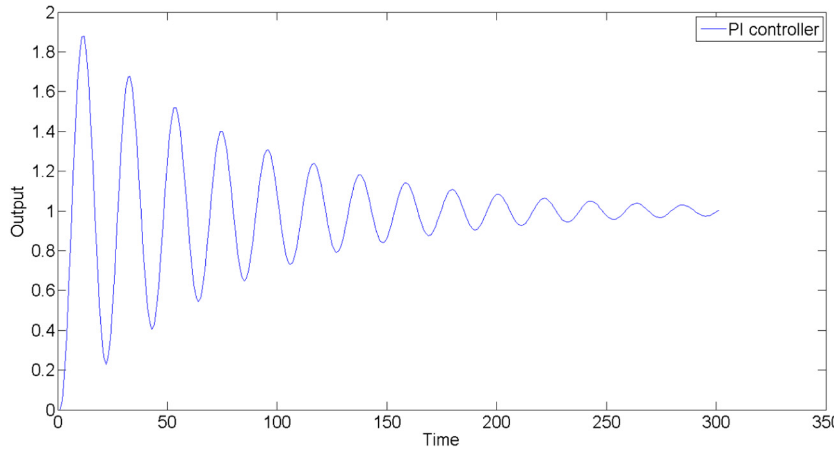


Figure 9 The unit step response for $K_p = 25$ and $K_d = 32.85$



Aircraft Engineering and Aerospace Technology: An International Journal
Volume 88 · Number 6 · 2016 · 799–809

Figure 10 The unit step response for $K_p = 50$ and $K_i = 10$



methods are used to find conventional controller gains different from the above studies. A fuzzy controller is created by the trial and error method. Integral of squared time multiplied by squared error is used as a performance function type in PSO:

$$ISTE = \int_0^{\infty} te^2(t)dt \tag{14}$$

Results and discussion

A simple roll angle automatic flight control system block diagram is given in Figure 1. Here φ is the roll rate output, e is the error vector, φ_{ref} is the reference input vector and δ_k is the roll plant input. One aircraft's stability derivatives are as follows (Mclean, 1990):

$$C_{L_{\delta_k}} = 0.014, \quad C_{L_p} = 0.3, S = 495 \ \ m^2, \quad b = 58.7\text{m},$$
$$I_x = 23.9 \text{ kg/m}^3$$

Using the stability derivatives, a roll plan can be obtained. The lateral movement of the stability derivatives using these terms as follows:

$$L_{\delta_k} = \frac{QSbC_{l_{\delta_k}}}{I_x} = 0.18, \ L_p = \frac{QSb^2C_{l_p}}{2I_x u_0} = 0.45 \tag{15}$$

Using the stability derivatives, the following expression is obtained as a transfer function of roll dynamic:

$$\frac{\varphi}{\delta_k} = \frac{L_{\delta_k}}{s(s\ L_p)} = \frac{0.18}{s(s + 0.45)} \tag{16}$$

The resulting eigenvalues of the open loop system are: $\lambda_1 = 0$, and $\lambda_2 = -0.45$. The aircraft is unstable. A transfer function is converted to the steady space of the system:

$$A = \begin{bmatrix} -0.45 & 0 \\ 1 & 0 \end{bmatrix}, \ B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ C = \begin{bmatrix} 0 & 0.18 \end{bmatrix} \tag{17}$$

Here, the states are β sideslip angle and r yaw rate. A feedback matrix K can be used to obtain stability:

$$K = \begin{bmatrix} 14.55 & 50 \end{bmatrix} \tag{18}$$

Using a feedback matrix, a stable A matrix is calculated as:

$$A^* = \begin{bmatrix} -15 & 50 \\ 1 & 0 \end{bmatrix} \tag{19}$$

The resulting eigenvalues of the open loop stable system are: $\lambda_1 = -10$, and $\lambda_2 = -5$. The aircraft is now stable. On the other hand, the new steady space of the system is converted to a transfer function:

$$\frac{\varphi}{\delta_k} = \frac{0.18}{S^2 + 15s + 50} \tag{20}$$

If the system has no sensor failures, the output and states are obtained as Figures 2 and 3. Here, a controller is not used. $F = \begin{bmatrix} -10 & 0 \\ 0 & -10 \end{bmatrix}$ is chosen in the observer equation. $T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is used. $G = \begin{bmatrix} 0 \\ 55.5556 \end{bmatrix}$ and $L = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ are obtained. $u = 1$ is used as a control input. Estimated states and residuals are obtained as shown in Figures 4 and 5. $r1$ and $r2$ show residuals of sideslip angle and yaw rate in Figure 5. If the system has a sensor fault, the fault is detected and isolated by using an observer. Here, a failure simulation prepared in "roll rate sensor" at iteration time = 20. $f_s = 1$ is used as a faulty vector. The fault is detected as in Figure 6. It can be seen that the roll rate has increased after 20 iterations. Residuals are obtained as shown in Figure 6.

Figure 11 Error functions

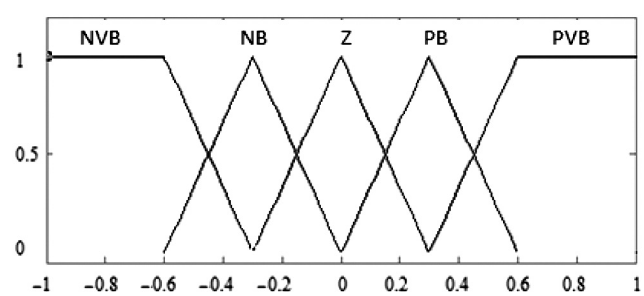


Figure 12 Derivative of the error functions

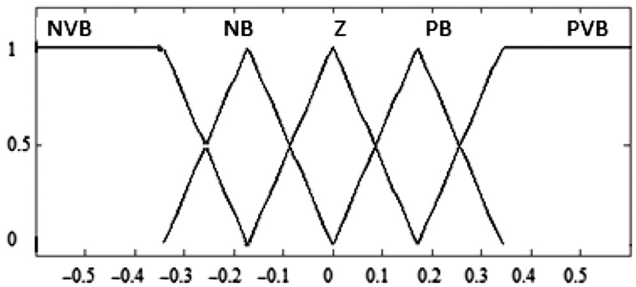


Figure 13 Output functions

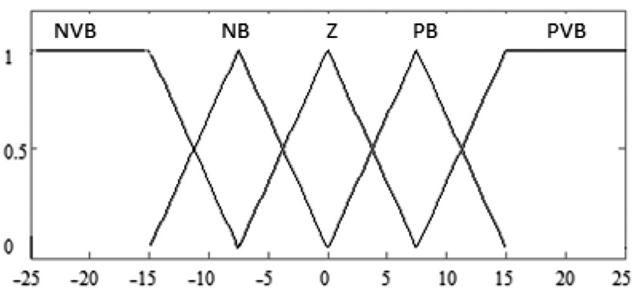


Table I The rules of table

	e				
ė	NVB	NB	Z	PB	PVB
NVB	NVB	NVB	NB	NB	Z
NB	NVB	NB	NB	Z	PB
Z	NB	NB	Z	PB	PB
PB	NB	Z	PB	PB	PVB
PVB	Z	PB	PB	PVB	PVB

By checking residuals, it can be seen that after a 20th iteration, *r*2 increases surpassing the threshold in Figure 7. Here, the fault is caused by a roll rate sensor malfunction. Accommodation can be achieved by switching. First, assuming that the controller is a P-type, $K_p = 1.257$ is obtained using the root-*locus* for equation (8). Using this gain value, the unit step response obtained is given in Figure 8. It appears to be around 20 per cent overshoot, and the desired output is obtained when Figure 8 is examined.

Second, the PD controller is considered to be designed for equation (8). In this case, the forward path transfer function and the closed loop transfer function are obtained as follows:

$$G(s) = \frac{0.18(K_p + K_d s)}{s(s + 0.45)},$$
$$\frac{\varphi(s)}{\varphi_{ref}(s)} = \frac{0.18(K_p + K_d s)}{s^2 + (0.45 + 0.18K_d)s + 0.18K_p} \quad (21)$$

Position, velocity and acceleration stability errors are as follows:

$$K_{ps} = \lim_{s\to 0} G(s) = \infty \text{ and } e_{ss} = \frac{1}{1 + K_{ps}} = 0 \quad (22)$$

$$K_v = \lim_{s\to 0} sG(s) = 0.4K_p \text{ and } e_{ss} = \frac{1}{K_v} = 2.5/K_p \quad (23)$$

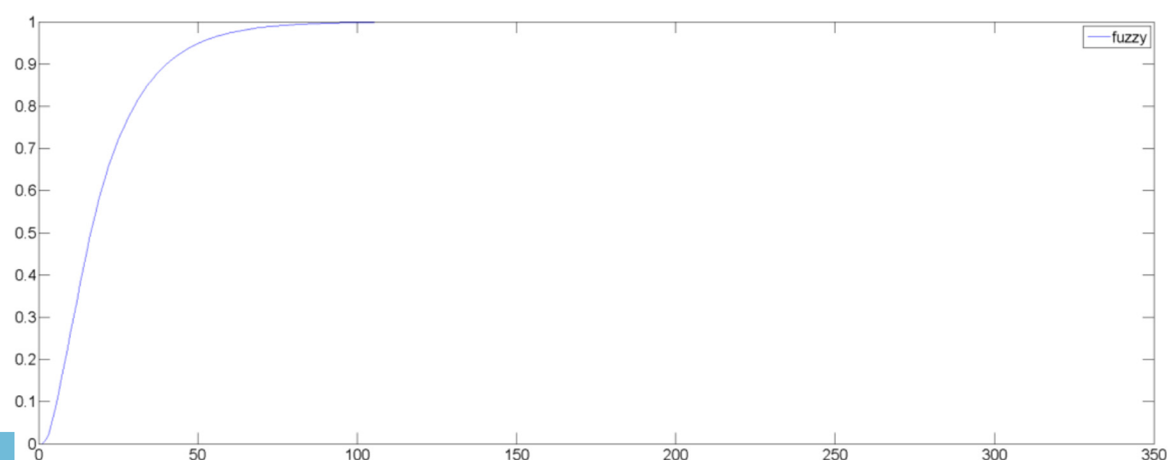
$$K_a = \lim_{s\to 0} s^2G(s) = 0 \text{ and } e_{ss} = \frac{1}{K_a} = \infty \quad (24)$$

The system is suitable for unit step input and ramp functions when equations (22)-(24) are examined. If the steady state failure is made smaller than 0.1, $K_p \geq 25$ is used. In this case, if the damping ratio is 0.707, $K_d = 32.85$ is obtained. Using these gain values, the unit step response obtained is given in Figure 9.

Third, the PI controller is considered to be designed for equation (8). In this case, the forward path transfer function and the closed loop transfer function are obtained as follows:

$$G(s) = \frac{0.18K_p(s + K_i/K_p)}{s^2(s + 0.45)},$$
$$\frac{\varphi(s)}{\varphi_{ref}(s)} = \frac{0.18(K_p + K_d s)}{s^2 + (0.45 + 0.18K_d)s + 0.18K_p} \quad (25)$$

Figure 14 The unit step response for fuzzy controller



Aircraft Engineering and Aerospace Technology: An International Journal
Volume 88 · Number 6 · 2016 · 799–809

Table II The algorithm code

Stages	Algorithm code
1	Initialize size of swarm, dimension of the problem, *c*1 and *c*2 coefficients, inertia weight. Size of swarm used is 50. Dimension of the problem is used as 2 $c_1 = c_2 = 1.494$ $w = 0.9$
2	Initialize the random functions. R_1 and R_2 are random functions in the range [0, 1]
3	Initialize velocities and positions. Here, initially the local best position is used
4	Determine initial population. Here, global best fitness and global best position are found. Here, the integral of squared time multiplied by squared error performance criterion formula is used. $ISTE = \int_0^\infty te^2(t)dt$
5	Update velocity [equation (13) is used]
6	Update swarm [equation (12) is used]
7	Update loop iterations to find global best position and global best fitness Find K_p and K_d
8	Run model using K_p and K_d

Position, velocity and acceleration stability errors are as follows:

$$K_{ps} = \lim_{s \to 0} G(s) = \infty \text{ and } e_{ss} = \frac{1}{1 + K_{ps}} = 0 \quad (26)$$

$$K_v = \lim_{s \to 0} sG(s) = \infty \text{ and } e_{ss} = \frac{1}{K_v} = 0 \quad (27)$$

$$K_a = \lim_{s \to 0} s^2G(s) = 0.4K_i \text{ and } e_{ss} = \frac{1}{K_a} = 2.5/K_i \quad (28)$$

If the Routh – Hurwitz test is performed, $0 < K_i/K_p < 0.45$ is obtained. Using suitable values ($K_p = 50$ and $K_i = 10$ are chosen), the unit step response obtained is given in Figure 10. It seems around a 90 per cent maximum overshoot, and the desired output will be obtained when Figure 10 is examined.

Fourth, the fuzzy controller is considered to be designed for equation (8). In the system, the error, derivative of the error and output membership functions are defined in Figures 11-13.

The rules of the table have been formed as shown in Table I. Using membership functions and rule table, the unit step response obtained is given in Figure 14. The algorithm code is presented in Table II. Using this algorithm, gains are calculated. $K_p = 3.8691$ and $K_d = 10.4906$ are calculated using the PSO algorithm for equation (8). It seems around a 3 per cent overshoot, and the desired output is obtained when Figure 15 is examined. The performance of each of the five controllers is given in Figure 16 for equation (8). A comparison of the methods is obtained in Table III. Here, calculated gains, max overshoots and settling times are shown. Error bands show a percentage of steps 0.1 per cent for t_{s_1}, 0.05 per cent for t_{s_2} and 1 per cent for t_{s_3} in Table III.

Conventional controllers (P controller, PD controller and PI controller), intelligent controller (fuzzy controller) and PD controller-based PSO are investigated. It can be seen that the P controller, PI controller and PD controller-based PSO cause overshoot. These overshoots are 18.5, 87.7 and 2.6 per cent, respectively. Overshoot is not seen using the PD controller and fuzzy controller. The steady state error is almost zero for all controllers. The effect of the PI controller will appear more efficient when the simulation runs for longer. The PD controller has the best settling time. The fuzzy controller is second best. The PD controller-based PSO is the third best, but the result is close to the other controllers.

Figure 15 The unit step response for PD controller based PSO

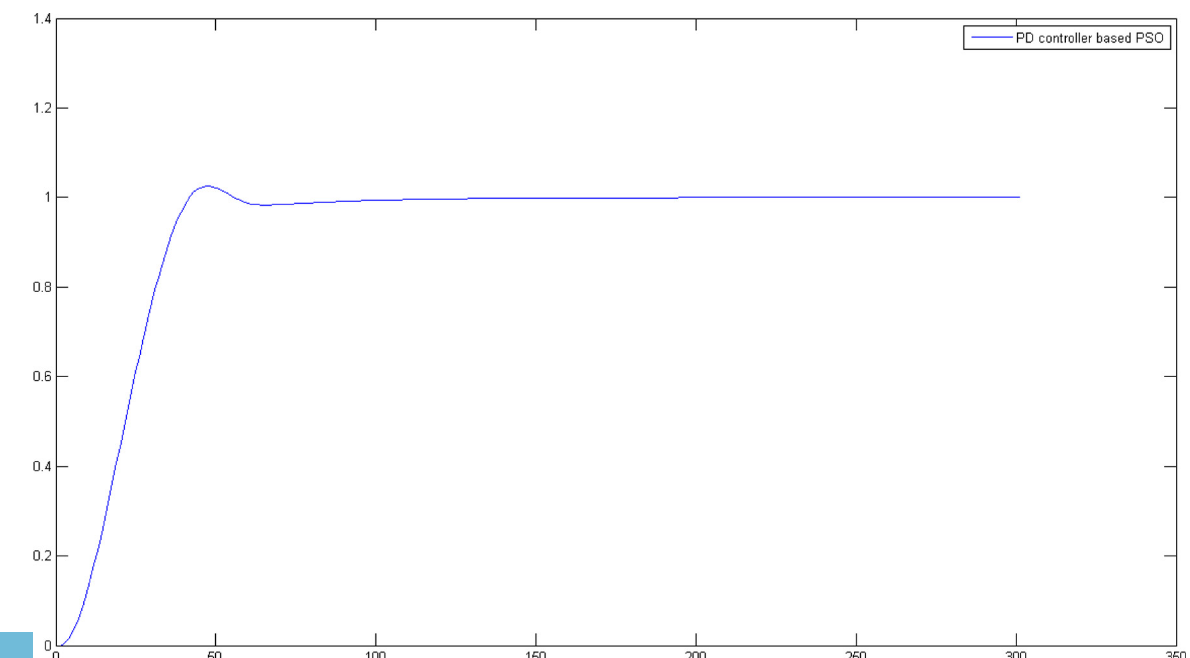


Figure 16 The unit step response for all controllers

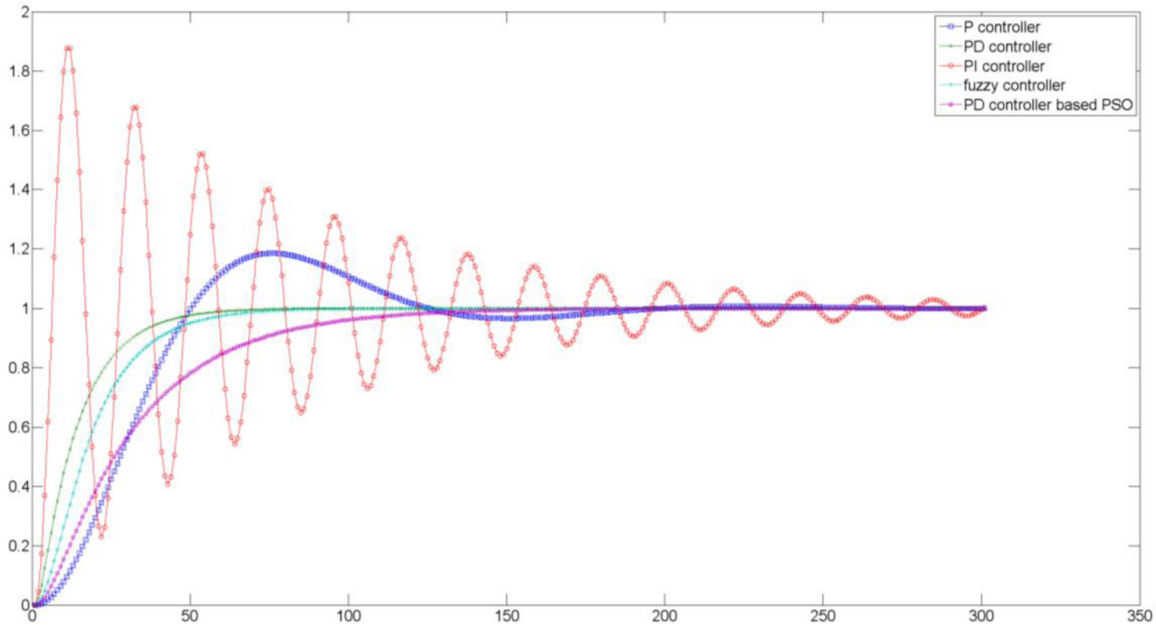


Table III Comparison of methods

Method	Coefficients	Maximum overshoot (%)	t_{s_1} (0.1%) (s)	t_{s_2} (0.05%) (s)	t_{s_3} (1%) (s)
P controller	$K_p = 1.257$	18.5	31.2	24.2	18.6
PD controller	$K_p = 25,\ K_d = 32.85$	–	9	7	6.1
PI controller	$K_p = 50,\ K_i = 10$	87.7	49.9	42.3	37
Fuzzy controller	–	–	10.8	8.5	7.4
PD controller-based PSO	$K_p = 3.8691,\ K_d = 10.4906$	2.6	20.5	15.9	14

Conclusion

In this study, first, faults on an aircraft model are detected and isolated. Second, a comparison of a P controller, PD controller, PI controller, fuzzy controller and PD controller-based PSO are realized utilizing the roll control system of an aircraft. The system is composed of a comparator, controller and aircraft equation of motion. Assuming that the sensor is an excellent measure, the transfer function of the feedback sensor is used as a unit function. The P-type controller is designed from a root *locus* plot. The PD type controller is obtained from desired stability errors using time domain. The PI type controller is designed using the Routh–Hurwitz stability criterion. In classical controllers, outputs having the preferred performance can be obtained by adjusting the gain, whereas for fuzzy controllers better values can be obtained by increasing the number of input and output functions and the number of rules. It can be seen that the PD controller-based PSO is the third best controller, but the system becomes more complex, and, therefore, traditional methods cannot be used. If PSO-based methods are used, much better results can be obtained. In this study, the PD controller-based PSO was not the best controller among the controllers. However, a traditional method provides us with the initial PID gain values for optimal tuning. Therefore, the benefit of using a PSO approach can be seen as a complementary solution for improving the performance of a PD controller designed by a conventional method.

References

Abaspour, A., Sadeghi, M. and Sadati, S. (2013), "Optimal fuzzy logic tuning of dynamic inversion flight controller", *Advance in Electronic and Electric Engineering*, Vol. 3 No. 1, pp. 7-24.

Alagoz, B., Ates, A. and Yeroglu, C. (2013), "Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control", *Mechatronics*, Vol. 23 No. 7, pp. 789-797.

Ammar, N. (2000), "Robust fault detection by simultaneous observers", *MSc*, Bilkent University, Ankara.

Bassi, S.J., Mishra, M.K. and Omizegba, E.E. (2011), "Automatic tuning of proportional-integral-derivative (PID) controller using particle swarm optimization (PSO) Algorithm", *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol. 2 No. 4, pp. 25-34.

Blanke, M., Lunze, J., Kinnaert, M. and Staroswiecki, M. (2006), *Diagnosis and Fault-Tolerant Control*, Springer-Verlag, Berlin, Heidelberg.

Caliskan, F. and Hajiyev, C. (2013), "A review of in-flight detection and identification of aircraft icing and

reconfigurable control", *Progress in Aerospace Sciences*, Vol. 60 No. 1, pp. 12-34.
Demirci, U. and Kerestecioglu, F. (2005), "Fault tolerant control with re-configuring sliding-mode schemes", *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 13 No. 1, pp. 175-187.
Duan, H., Yu, Y. and Zhao, Z. (2013), "Parameters identification of UCAV flight control system based on predator-prey particle swarm optimization", *Science China Information Sciences*, Vol. 56 No. 1, pp. 1-12.
El-Saady, G., Ibrahim, E.N.A. and Raafat, A.A. (2014), "Automatic landing system design using PSO method for Charlie aircraft", *International Journal of Engineering and Technical Research (IJETR)*, Vol. 2 No. 10, pp. 45-49.
Hajiyev, C. and Caliskan, F. (2005), "Sensor and control surface/actuator failure detection and isolation applied to F-16 flight dynamic", *Aircraft Engineering and Aerospace Technology*, Vol. 77 No. 2, pp. 152-160.
Hassan, L., Sadati, S.H. and Karimi, J. (2013), "An optimal fuzzy logic guidance law using particle swarm optimization", *International Journal of Computer Applications*, Vol. 69 No. 3, pp. 40-47.
Jingping, S., Weiguo, Z. and Suilao, L. (2011), "A reconfigurable control method based on face searching algorithm for aircraft with multiple effectors", *Procedia Engineering*, Vol. 15 No. 1, pp. 1251-1255.
Karasakal, O., Yeşıl, E., Güzelkaya, M. and Eksın, İ. (2005), "Implementation of a new self-tuning Fuzzy PID controller on PLC", *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 13 No. 2, pp. 277-286.
Kennedy, J. and Eberhart, R.C. (1995), "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks (ICNN)*, *Perth*, Vol. 4, pp. 1942-1948.
Khan, A.H., Weiguo, Z., Jingping, S. and Khan, Z.H. (2011), "Optimized reconfigurable modular flight control design using swarm intelligence", *Procedia Engineering*, Vol. 24 No. 1, pp. 621-628.
Khan, A.H., Weiguo, Z., Khan, Z.H. and Jingping, S. (2012), "Evolutionary computing based modular control design for aircraft with redundant effectors", *Procedia Engineering*, Vol. 29 No. 1, pp. 110-117.
Mclean, D. (1990), *Automatic Flight Control Systems*, Prentice Hall, Upper Saddle River, NJ.
Meng, W., Wei, C. and Tang, J. (2011), "Cultural algorithm based on particle swarm optimization for clearance of flight control law", *6th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, *Beijing*, pp. 418-423.
Meng, Z., Weiguo, Z., Yong, S. and Ning, N. (2010), "Reconfigurable flight control system design using eigenstructure assignment and particle swarm optimization algorithm", *2nd International Conference on Advanced Computer Control (ICACC 2010)*, Piscataway, NJ, Vol. 3, pp. 362-366.
Min, B.M., Shin, H.S., Tahk, M.J. (2006), "Control system design for an autonomous helicopter using particle swarm optimization", *25th International Congress of the Aeronautical Sciences (ICAS)*, *Hamburg*.
Montazeri-Gh, M., Jafari, S. and Nasiri, M. (2013), "Application of particle swarm optimization in gain tuning of integrated flight and propulsion control", *International Journal of Aerospace Sciences*, Vol. 2 No. 3, pp. 55-70.
Nelson, R.C. (1998), *Flight Stability and Automatic Control*, WCB/McGraw-Hill, Boston, MA.
Ömürlü, V. and Yildiz, İ. (2011), "Self-tuning fuzzy PD-based stiffness controller of a 3 × 3 Stewart platform as a man-machine interface", *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 19 No. 5, pp. 743-752.
Oner, K., Cetinsoy, E., Sirimoglu, E., Hancer, C., Unel, M., Aksit, M., Gulez, K. and Kandemir, I. (2012), "Mathematical modeling and vertical flight control of a tilt-wing UAV", *Turkish Journal Electrical Engineering and Computer Sciences*, Vol. 20 No. 1, pp. 149-157.
Romero, J., Sanchis, R. and Balaguer, P. (2011), "PI and PID auto-tuning procedure based on simplified single parameter optimization", *Journal of Process Control*, Vol. 21 No. 6, pp. 840-851.
Shi, Y. and Eberhart, R.C. (1998), "A modified particle swarm optimizer", *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, IEEE Press, Piscataway, NJ.
Solak, E. (2001), "Observability and observers for nonlinear and switching systems", PhD, Bilkent University, Ankara.
Stevens, B. and Lewis, F. (2003), *Aircraft Control and Simulation*, John Wiley Sons, Hoboken, NJ.
Wenyue, M., Chen, W. and Jie, T. (2011), "Cultural algorithm based on particle swarm optimization for clearance of flight control law", *6th IEEE Conference on Industrial Electronics and Applications*, Piscataway, NJ, pp. 418-423.

About the author

Emre Kiyak was born in 1978. He received the MSc degree (2003) from Anadolu University, Turkey, in Civil Aviation, and he received the PhD degree (2008) from Anadolu University in Civil Aviation. He is working as an Assistant Professor at the same university since 2008. His research interest includes fault tolerant control, flight control systems, fuzzy logic. Emre Kiyak can be contacted at: ekiyak@anadolu.edu.tr