Robotica (2011) volume 29, pp. 929–938. © Cambridge University Press 2011
doi:10.1017/S0263574711000154

PID and inverse-model-based control of a twin rotor system

S. F. Toha†* and M. O. Tokhi‡

†*Department of Mechatronics, Faculty of Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia*
‡*Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK*

(Received in Final Form: February 8, 2011. First published online: March 15, 2011)

SUMMARY

The use of active control techniques has intensified in various control applications, particularly in the field of aircraft systems. This paper presents an investigation into the control of rigid-body and flexible motion of a twin rotor multi-input multi-output system (TRMS) using intelligent inverse-model-based control schemes. The TRMS is an aerodynamic test rig representing the control challenges of modern air vehicle. The augmented feedback PID and feedforward inverse-model-based control has led to good tracking response and vibration reduction of the TRMS, with the use of particle swarm optimisation (PSO). As a comparison, methods using PID controllers are also presented. Experimental results are obtained using the test rig, confirming the viability and effectiveness of the proposed methodology as opposed to conventional PID controllers. The results and evidence from this method are justified, presented and discussed.

KEYWORDS: Twin rotor system; PID; Inverse model; Particle swarm optimisation.

1. Introduction

PID control was first introduced by Callender in 1936.[1] The controller applies a signal to the process that is proportional to the actuating signal, on top of adding the integral and the derivative of the actuating signal. Since its introduction, conventional PID control has been widely used for example in chemical and paper industries, where about 97% of regulator controllers have the PID structure.[2] This could be due to their low cost, inexpensive maintenance, as well as simplicity in controller design and operation.[3,4] The widespread use of this controller has led to research towards finding optimal values of the PID controller's parameters. Ziegler and Nichols proposed the classical tuning rules for the PID controller based on trial and error.[5] Moreover, a variety of tuning methods have been used to improve the performance of PID controllers, such as Cohen and Coon,[6] pole placement[7] and gain-phase margin method.[8] Unfortunately, it has been difficult to tune PID controller gains precisely because many industrial plants are often very complex consisting of issues such as higher order, time delays and non-linearities.[9] Some disadvantages of the classical techniques for tuning PID controllers are (i) excessive number of rules to set the gains, (ii) inadequate dynamics of closed-loop responses, (iii) difficulty to deal with non-linear processes and (iv) mathematical complexity of the control design.[10] Moreover, Van Overschee and De Moor[11] have reported that 80% of PID controllers are poorly/less optimally tuned and operate inappropriately,[12,13] followed by another 30% of the PID loops operating in the manual mode and 25% of PID loops actually operating under default factory settings. This shows that the determination of PID controller parameters is a troublesome problem in many industrial applications.[14] To overcome these difficulties, various types of modified conventional PID controllers such as autotuning[15] and intelligent PID controllers[16] have been developed.

Recent developments in the area of intelligent optimisations have attracted many researchers to utilise particle swarm optimisation (PSO) in order to improve the traditional methods in tuning the PID controller parameters.[17,18] PSO was first introduced by Kennedy and Eberhart in 1995.[19] It is a nature-inspired population-based optimisation technique based on simulating the phenomenon that takes place with flock of birds and school of fish. As the name suggests, PSO operates on a swarm of particles, where each particle represents a candidate solution to the problem. The swarm is flown through the search space, and each particle is attracted to both the best position encountered by itself so far, as well as the global best position found by its neighbourhood. The particles evaluate, in each iteration, their positions relative to the global fitness and use their memorised best positions to adjust their own velocities and subsequent positions. In this way, the particles tend to fly towards better and better searching areas over the search space. As a result of such social dynamics, a standard swarm normally converges to a small area around the optimum solution.[20] See ref. [21] for a detailed description of the PSO algorithm.

The vast majority of conventional control techniques have been devised for linear time-invariant systems that are assumed to be completely known and well understood. In most practical instances, however, the system to be controlled is non-linear, time-varying and the basic physical processes in it are not completely known *a priori*. These types of model uncertainties are extremely difficult to manage, even with conventional adaptive techniques. The need for control methods for the aforementioned processes becomes very important. Therefore, this paper presents a comparative study of classical PID control and intelligent inverse-model-based control scheme within a twin rotor system.

* Corresponding author. E-mail: tsfauziah@iium.edu.my

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

The parametric model of the twin rotor multi-input multi-output system (TRMS) has previously been reported, and it has been shown that the system model is accurate and stable.[22,23] Therefore, the same system identification technique is employed to obtain the inverse model of the system. The inverse models of dynamic systems play a crucial role in many control strategies.[24] Inverse dynamics identification is defined as finding the inverse mapping of a system, where it is useful to know the inverse dynamics of a plant in order to implement a good control scheme. The aim of dynamic model inversion is to cancel the flexural effects of the controlled plant by constructing its inverse mapping and using it in the control law. This makes it feasible to employ linear control system tools for achieving the desired control objectives.[25]

Myriad methodologies have been scrutinized in terms of obtaining inverse model control. For example, Danai[26] has used a feedforward network as the inverse model of the effect of the blade adjustment on helicopter vibration. This method includes priory knowledge of the process by defining the initial coefficients of the internal model. Zeng and Zhu[24] on the other hand designed a dynamic inversion controller for both altitude loop and attitude loop from a simplified mathematical model derived from generic YAMAHA R-50 unmanned helicopter simulation model. Rahideh *et al.*[25] have developed model inversion control in combination with genetic algorithm (GA) to tune a proportional-derivative (PD) controller. A neural network element is then integrated with a feedback control system to compensate for model inversion error. However, designing an effective inverse model control presented in the literature requires *a priori* knowledge of the actual system parameters such as structural dimensions, inertias and cross-coupling effect which may not be available or accurately known for a practical system.

Therefore, the current work addresses the use of parametric approach by utilising the input and output data pairs of the real TRMS in order to develop the inverse model of the system. The twin rotor system used in this work is described in Section 2. Section 3 describes the dynamic PSO with spread factor (PSO-SF) technique used for both inverse model and PID control optimisation. Section 4 describes feedback PID control. Inverse-model-based control approach is described in Sections 5 and 6, while Section 7 presents a comparative assessment of feedback PID control and the proposed inverse-model-based control scheme. Concluding remarks are given in Section 8.

2. The Twin Rotor System

The TRMS is a laboratory set-up developed by Feedback Instruments Limited.[27] Although the TRMS does not fly, its behaviour in certain aspects resembles that of a helicopter. For example, like a helicopter it possesses a strong cross-coupling between the collective (main rotor) and the tail rotor. In a typical helicopter, the aerodynamic force is controlled by changing the angle of attack of the blades. However, in the TRMS the aerodynamic force is controlled by varying the speed of the motors. It is driven by two DC motors at both ends of the beam which are main rotor motor and tail rotor motor. Its two propellers are perpendicular to each other and joined by a beam pivoted on its base that can rotate freely in the horizontal and vertical planes. The beam can thus be moved by changing the input voltage in order to control the rotational speed of the propellers. The articulated joint allows the beam to rotate in such a way that its ends move on spherical surfaces. The system is equipped with a pendulum counterweight hanging from the beam, which is used for balancing the angular momentum. Table I lists some characteristic parameters of the TRMS used in this work, and a schematic diagram of the TRMS is shown in Fig. 1. The X-axis corresponds to yaw axis, Y-axis corresponds to pitch axis and Z-axis corresponds to the third orthogonal axis. Therefore, the horizontal plane becomes YZ-plane and the vertical plane becomes XZ-plane.

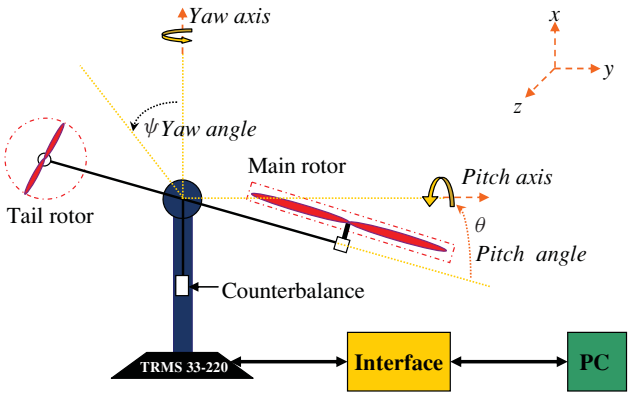


Fig. 1. Twin rotor multi-input multi-output system.

Table I. Characteristic parameters of the TRMS.

Parameters	Value
Maximum input voltage range	+/−10 V
Length of the beam	0.49 m
Mass of the main DC motor with main rotor	0.228 kg
Mass of the tail DC motor with tail rotor	0.206 kg

The system is balanced in such a way that when the motors are switched off, the main rotor end of the beam is lowered. The controls of the twin rotor system are the supply voltages of the motors. It is important to note that the geometrical shapes of the propellers are not symmetric. Accordingly, the system behaviour in one direction is different from that in the other direction. Rotation of a propeller produces an angular momentum which, according to the law of conservation of angular momentum, is compensated by the remaining body of the TRMS beam. This results in interaction between the moment of inertia of the motors with propellers. This interaction directly influences the velocities of the beam in both vertical and horizontal planes. The system is interfaced with a personal computer through a data acquisition board, PCL-812PG. The measured signals are position of the beam, which constitutes two position angles, and the angular velocities of the rotors. Angular velocities of the beam are obtained through real time kernel (RTK) software by differentiating and filtering the measured position angles of the beam. RTK provides a mechanism of real-time measurements and control of the TRMS in the WINDOWS environment.

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

During the experimental procedure, the beam motion is allowed unrestricted in the pitch and the yaw planes, so that the motion of TRMS includes all the dynamic aspects of the system. The input and output from the system are main rotor input, $u_v(t)$; tail rotor input, $u_h(t)$; main rotor output, $y_v(t)$; and tail rotor output, $y_h(t)$. It has been proved in previous studies that the pitch plane has faster dynamics than the yaw plane. This exhibits a strong coupling from (i) $u_v(t)$ to $y_v(t)$, (ii) $u_v(t)$ to $y_h(t)$, (iii) $u_h(t)$ to $y_h(t)$ and no interaction between $u_h(t)$ and $y_v(t)$ at the lower frequency region.[28] This strong coupling is the main feature of the twin rotor system rather than of rotorcraft in general. As there is no strong coupling between $u_h(t)$ and $y_v(t)$ which is the horizontal motion of the plane, the main interest in this work is in the hovering property (motion in the vertical plane) of the TRMS. Thus, the measurement taken comprises the input and output data pairs of the main rotor (vertical motion) of the system.

3. Particle Swarm Optimisation

In the PSO algorithm, each individual named as a particle represents a potential solution to a problem. Each particle is moving and hence has a velocity. Also, each particle remembers the position it was in and where it had its best result so far. Moreover, the particles in the swarm cooperate and exchange information about what they have discovered within the search space they have visited. A fitness function is the metric determining the solution's optimality. The fitness of each particle is evaluated in each iteration. The position of the particle with the best fitness, called the global best, is preserved in the memory. Each particle also preserves in memory the best position and its best fitness, called the particle best. These best fitness positions influence how the solutions change or particles move in the search space. The PSO algorithm produces a random population initially and generates the next population based on current cost, but unlike GAs, it does not need reproduction or mutation to produce the next generation. Thus, PSO is faster in finding solutions compared with evolutionary computation techniques.

Dynamic PSO-SF is employed in this paper.[22,29] The algorithm is found to be highly effective in improving major issues in simple PSO, which has a tendency to converge to local optima, mainly due to lack of diversity in the particles as the algorithm proceeds and improper selection of other parameters. The mathematical representation of PSO-SF is given as

$$x_{id_{new}} = x_{id} + v_{id_{new}}, \tag{1}$$

$$v_{id_{new}} = (w \times v_{id}) + c_1(rand_1(p_{id} - x_{id})) + c_2(rand_2(p_{gd} - x_{id})), \tag{2}$$

with w, c_1 and c_2 given as

$$w = \exp(-iter/(spread_factor \times \max_iteration)), \tag{3}$$

$$spread_factor = 0.5(spread + deviation), \tag{4}$$

$$c_1 = 2(1 - iter/\max_iteration), \tag{5}$$

$$c_2 = 2, \tag{6}$$

where x_{id} and v_{id} represent the position vector and velocity vector of the ith particle in the d-dimensional search space, respectively; c_1 and c_2 are the acceleration coefficients and $rand_1$ and $rand_2$ are two separate random numbers generated uniformly in the range of [0, 1], and are responsible for providing randomness to the particles of the swarm. The velocity vector in Eq. (2) is used to calculate particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (p_{id}) and the global best experience (p_{gd}). Then, the particle flies towards a new position according to Eq. (1).

In order for particles to keep exploring the search space, it is imperative that they must know their whereabouts and relative distances from each other. The spread factor in PSO algorithm measures the distribution of particles in the search space as well as the precision and accuracy of the particles with respect to global optimum. Therefore, when all the particles move within the vicinity of global optimum, both the dynamic SF and hence the inertia weight will drop in value drastically. This will not only force all the particles to converge but also allow the algorithm to achieve extremely high precision. Suitable selection of the inertia weight provides a balance between global exploration and local exploitation and results in less iteration on average to find a sufficiently optimal solution. In conclusion, a large inertia weight encourages global exploration, while a small value promotes local exploitation. It is crucial that there is high diversity in the early stages of the search, whereas low diversity is preferable in later stages. Therefore, a suitable value for w provides the desired balance between the global and local exploitation ability of the swarm and, consequently, improves the effectiveness of the algorithm.[30]

Throughout this work, PSO-SF is used in search for optimal parameter values of the PID controllers as well as parameter values of inverse model for the twin rotor system. The PSO-SF process is best explained in Fig. 2. The PSO-SF parameter optimisation embraced in this work minimises the objective function $F(\bullet)$ defined as mean squared error (MSE), which is the error between the actual output, $y(t)$, and the predicted output, $\hat{y}(t)$, of the system.

$$F = MSE = \frac{1}{n}\sum_{i=1}^{n}(error)^2, \tag{7}$$

where n is the length of data.

4. Feedback PID Control of TRMS

A set of PD, PI and PID control schemes are implemented and tested within the simulation environment of the TRMS platform and the corresponding results are presented in this section. It has been shown in a previous work that the parametric model obtained is stable and accurate.[22,23] Therefore, the control scheme is developed to obtain a good input tracking for the hovering motion of the system. The model transfer function of the TRMS in the vertical motion is given as

$$H_v(z) = \frac{-0.7428z^3 - 0.5711z^2 + 0.3548z + 0.7871}{z^4 - 0.0049z^3 - 0.0018z^2 - 0.0015z + 0.0019} \tag{8}$$

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

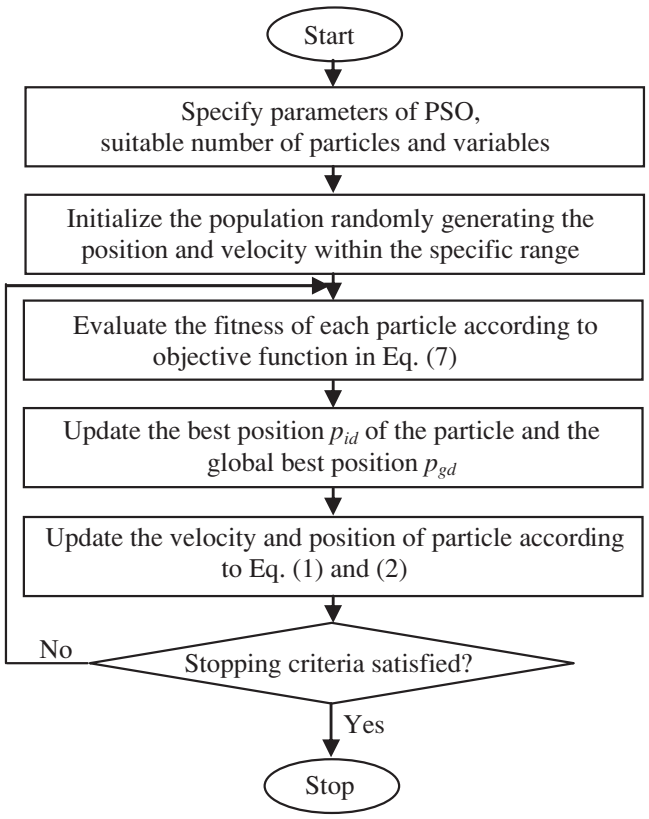


Fig. 2. The flowchart of PSO-SF algorithm.

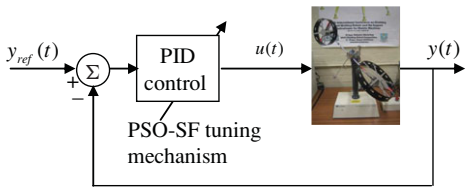


Fig. 3. PID controller with PSO-SF tuning mechanism for the TRMS.

The conventional closed-loop control method determines the gain at which a system with proportional control may diverge and then derives the control gain and derivative values from the gain wherein the oscillations are contained. The PID control is given as follows:

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s, \tag{9}$$

where K_P, K_I and K_D are the proportional, integral and derivative gains of the controller, respectively. Therefore, the PSO-SF optimisation method can provide parameter tuning while reducing oscillation in a closed-loop environment as described in Fig. 3.

Table II shows controller parameters obtained at the end of running the PSO-SF for 100 iterations with a swarm size of 50 particles. It is noted in Table III that the optimisation for PI controller converged faster with 42 iterations as compared with PD and PID controller. On the other hand, even though PID controller optimisation converged about twice slower than PD controller, it achieved the smallest value of MSE, and the execution time was still comparable. This

Table II. Tabulated PD, PI and PID controller parameters with 100 iterations.

Controller	Controller parameters		
	K_P	K_I	K_D
PD	5.068	–	1.473
PI	6.368	2.532	–
PID	9.618	7.425	4.053

Table III. Performance of PD, PI and PID controller with 100 iterations.

Controller type	Mean squared error	No. of iterations	CPU execution time (s)
PD	1.2441×10^{-9}	75	3.1185
PI	1.1199×10^{-10}	42	3.1102
PID	4.3088×10^{-11}	84	3.1591

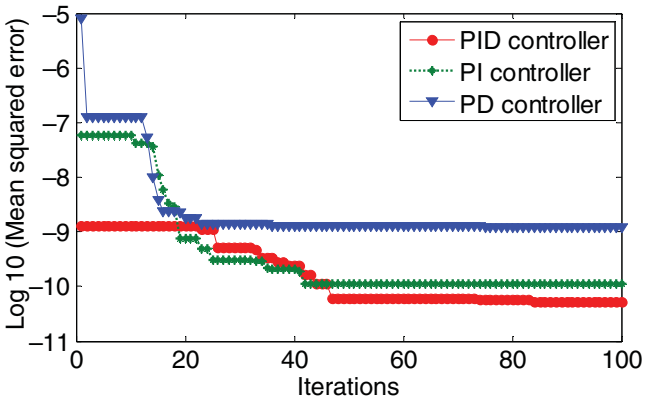


Fig. 4. Convergence graph of PSO-SF optimisation.

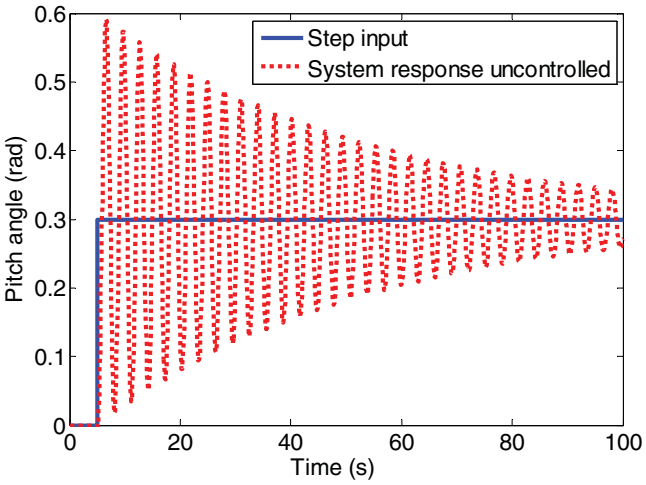


Fig. 5. Open-loop response of the TRMS to a reference step input.

is further illustrated in Fig. 4, which shows the convergence graph of PSO-SF optimisation as a function of number of iterations. The results thus show that PID control can achieve good convergence with the smallest MSE, value and within acceptable execution time.

Figure 5 shows the open-loop response of the system to a step input voltage to the main rotor, $u(t)$. It is noted

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

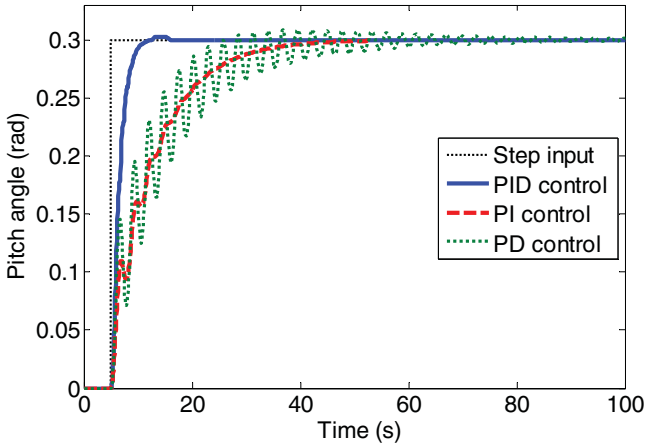


Fig. 6. System response with PD, PI and PID controllers using PSO-SF tuning.

that the system response is highly oscillatory due to the system dynamics. Figure 6 shows the system response with the developed controllers in Table II. The obtained results reveal that the control scheme was able to reduce the system vibration while maintaining input tracking performance of the TRMS rig. The system performance with a feedback controller was significantly improved when the feedback control component was added. It is evident from the output response that the system performance with PD control was poor, and the tracking performance as well as the rise time were improved with PI control even though some ripple was noted around the rigid-body motion of the system. The PID control was far superior to the other controllers whereby the system response was characterised by smooth tracking with small overshoot, shorter settling time with minimal residual vibration. The settling time of the response with PID may still be improved. The anticipated improvement in the settling time as well as reduction of oscillation in output response is shown in Section 5.

5. Inverse-Model-Based Control

In this section, the identification of the inverse model of the TRMS is discussed. The forward parametric model of the system that mimics the TRMS behaviour has been obtained in the previous work using the same PSO-SF optimisation approach.[22,23] Dynamic model inversion is a popular method to achieve feedback linearization because it is easy to apply and its physical meaning is clear.[31] The inverse model is designed to replicate the inverse of the system behaviour. The input of the inverted model is pitch angle of rotor, $y(t)$, and the output is the main voltage to the system, $u(t)$. Figure 7 depicts a schematic diagram of the inverse model identification technique where PSO-SF is used to obtain the inverse model parameters of the system. The predicted input using pseudo-random binary signal (PRBS) for the inverse-model-based scheme is presented against the actual input in Fig. 8. According to Ljung,[32] the type of input signal to be used should be able to excite all the dynamic modes of interest where the spectral content of the input signal should be rich in frequency corresponding to system bandwidth. It

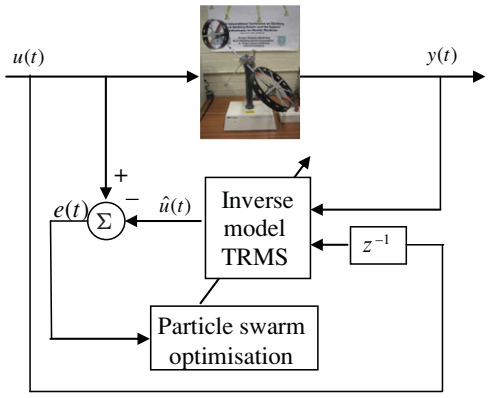


Fig. 7. Schematic diagram of the inverse model identification.

Table IV. The PSO-SF parameters for inverse model identification.

Characteristics	
1. Number of iterations	300 iterations
2. Number of particles	50 particles
3. Number of variables	8 variables

should also be rich in amplitude level whereby it should have different levels of input amplitudes over the whole range of operation. This goal can be achieved by exciting the process with PRBS input signal.

The characteristic parameters of the PSO-SF used in the inverse modelling identification are shown in Table IV. The difference between the predicted and actual input is recorded as error, $e(t) = u(t) - \hat{u}(t)$, which in turn is used to form the objective function of the optimisation process, as in Eq. (7).

The convergence graph of inverse model identification using PSO-SF is shown in Fig. 9. The maximum iteration was set to 300 with 50 particles according to Table III. It is noted that PSO-SF inverse model identification achieved an accurate inverse model within 49.907 s execution time. It is further shown that PSO-SF has the ability to converge faster with small MSE value, 1.496×10^{-14} in only 149 iterations. The algorithm thus gave stable results through termination. The maximum number of iterations was used as the termination criterion.

The inverse model transfer function of the TRMS in the vertical motion obtained by using the PSO-SF optimisation is given as

$$H_{Inv}(z) = \frac{0.8362z^3 + 0.7375z^2 - 0.2259z - 0.4501}{z^4 - 0.0037z^3 - 0.00036z^2 + 0.0018z + 0.0025} \quad (10)$$

The obtained inverse model of the system was then placed in a cascade with the parametric forward model of TRMS as shown in Fig. 10. The simplest approach for controller design is a completely open-loop control strategy, in which the controller is the inverse of the process. For this open-loop inverse-model control, the input to the inverse model is the desired value of pitch angle, $y_{ref}(t)$; the output of the inverse model, that is the input to the plant, is the voltage of the main rotor, $u(t)$; and the output of the plant is the pitch angle, $y(t)$.

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

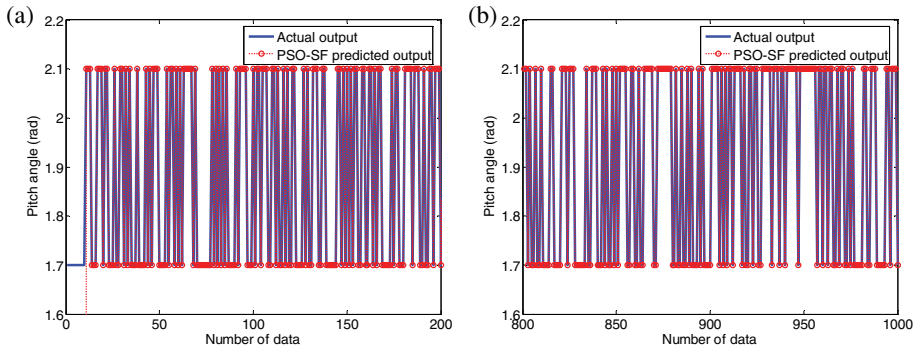


Fig. 8. Actual and PSO-SF predicted inverse model output of the system (a) First 200 data, (b) Last 200 data.

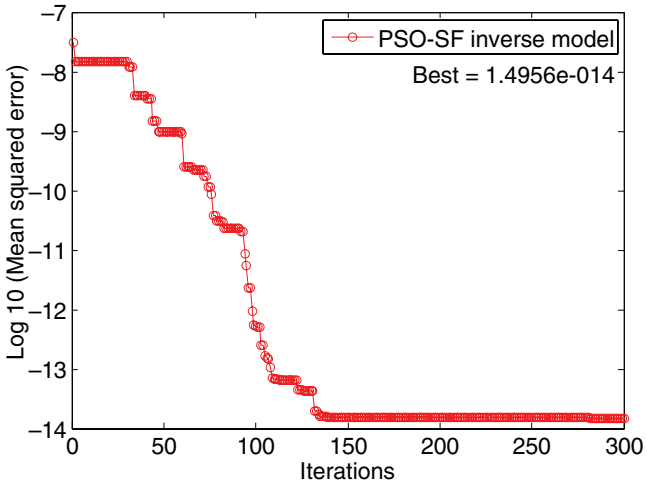


Fig. 9. Convergence graph of PSO-SF inverse model identification.

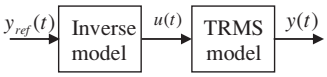


Fig. 10. The open-loop inverse-model control.

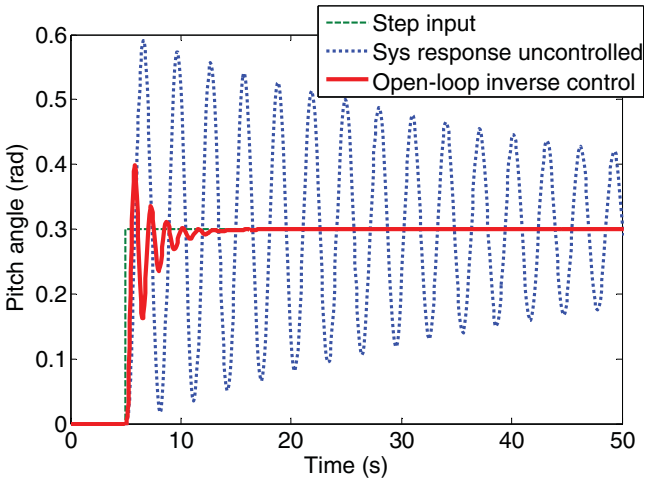


Fig. 11. The open-loop response of model and inverse model control.

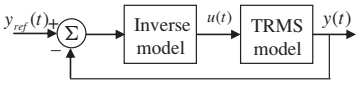


Fig. 12. Block diagram of a direct inverse-model control system.

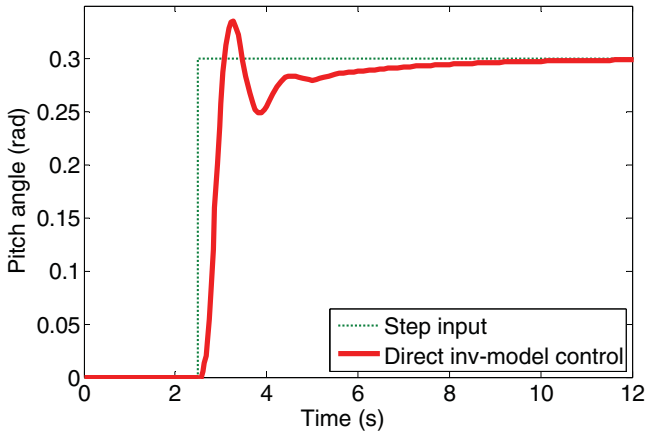


Fig. 13. Closed-loop response of the direct inverse-model control to a step reference input.

The output response of the open-loop inverse-model control in Fig. 11 shows that the inverse model has reduced system vibration to some extent, around the set point (rigid-body motion). This shows that the inverse model technique can be used to enhance the tracking characteristics of the system.

6. Inverse-Model-Based Control Approaches

6.1. Direct inverse-model control

In this section, the potential of inverse model control is investigated in closed-loop configurations for set-point tracking and vibration control in the TRMS. After the promising results from the open-loop inverse-model control, direct inverse-model control was developed and investigated in a unity feedback closed-loop configuration as depicted in Fig. 12. The corresponding result achieved is shown in Fig. 13. It is noted that the response was faster compared with the PD, PI and PID controls in Fig. 6. The oscillations in the system response were also reduced compared with the open-loop control response in Fig. 10.

6.2. Augmented PID and feedforward inverse-model control

In this work, an augmented PID and feedforward inverse-model control is developed for control of rigid-body motion of the TRMS system to enhance the tracking performance

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

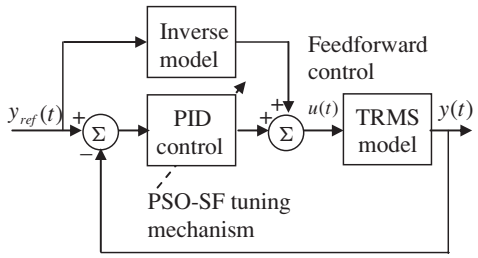


Fig. 14. Block diagram of the augmented PID and feedforward inverse-model control scheme.

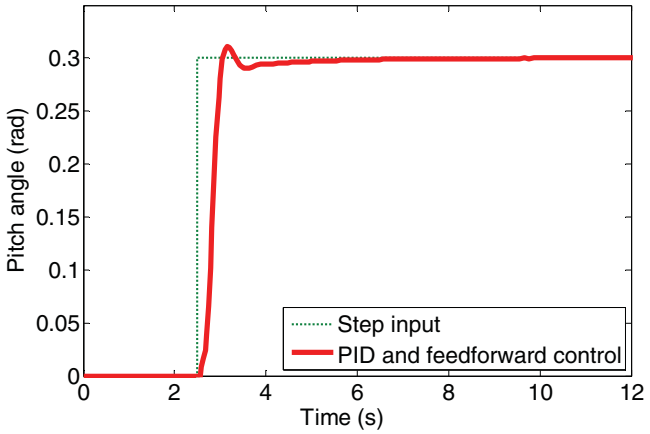


Fig. 15. Closed-loop response of the augmented PID and feedforward inverse-model control to a step reference input.

Table V. The characteristic parameters of the PSO-SF used for PID controller optimisation.

Characteristics	
1. Number of generations	100 generations
2. Number of particles	50 particles
3. Number of variables (K_P, K_I, K_D)	3 variables

of the TRMS in hovering position. The control structure is shown in Fig. 14.

The total input that enters the plant is the sum of the feedback control signal and the feedforward control signal calculated from the inverse dynamic model. The model uses the desired pitch angle as the input and the output is the main voltage. In this configuration, the PID controller further helps with improving the set-point tracking. As noted in Fig. 15, while the inverse model acts in the same manner as in Fig. 13, the output response of augmented PID and feedforward inverse-model control system is better than that of direct inverse-model control system. The overshoot was reduced by 14%, and the settling time was reduced to 6 s. The optimum values of PID parameters were obtained using the PSO-SF. The characteristic parameters of the PSO-SF used are listed in Table V.

6.3. Augmented PID and feedback inverse-model control

The combination of feedback inverse-model control loop for flexible motion control (Loop 2) and PID control for rigid-body motion control (Loop 1) is devised as shown in Fig. 16. The two control loops are decoupled by using a highpass filter

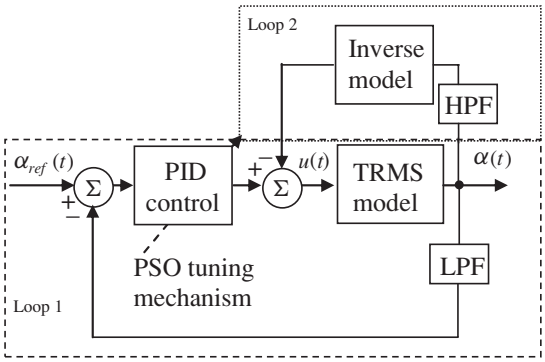


Fig. 16. Block diagram of augmented PID and feedback control scheme.

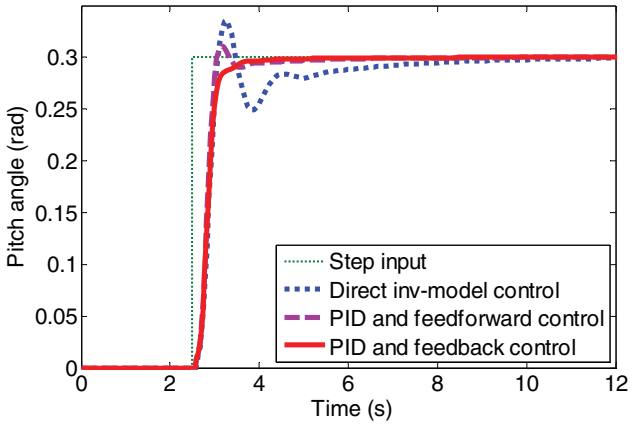


Fig. 17. Closed-loop response of the control systems.

Table VI. PID parameters for augmented PID and feedforward control and augmented PID and feedback control.

Augmented PID and feedforward control			Augmented PID and feedback control		
K_P	KI	K_D	K_P	K_I	K_D
12.36	7.58	5.23	1.73	1.21	0.47

(HPF) in Loop 2 and a lowpass filter (LPF) in Loop 1. The cut-off frequencies of both HPF and LPF were set to 0.15 Hz. From previous study,[22,23] the frequency of the main dominant resonance mode of the TRMS is at about 0.349 Hz. The main purpose of the LPF in the Loop 1 is to reject frequencies above 0.15 Hz including the main resonance mode of the system so that the first loop will solely concentrate on control of the rigid-body motion of the system. The Loop 2 is a negative feedback loop where the inverse model is used as a controller to augment the hovering control of the system albeit of using the PID feedback controller alone. The HPF will reject the low-frequency (rigid-body) dynamics of the system. Thus, this loop will control the flexible motion dynamics (vibration) of the system. The optimum PID parameter values obtained using PSO-SF tuning for augmented PID and feedback control are given in Table VI.

Figure 17 shows a comparison of system performance with the three inverse-model-based control schemes. It is evident from the output responses that the direct inverse-model control was relatively poor in performance and

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

Table VII. Performances of the employed control schemes in terms of time-domain response parameters.

Control strategy	Time-domain parameters			
	Rise time (s)	Overshoot (%)	Settling time (s)	Steady-state error (%)
1. Open loop	0.492	97.349	44.628	−30.723
2. PD control	11.791	2.825	39.480	1.606
3. PI control	17.203	0	26.072	−0.269
4. PID control	9.327	0.768	31.243	
5. Open-loop inverse-model control	0.563	8.653	28.397	−7.253
6. Direct inverse-model control	3.866	3.469	14.685	−0.417
7. Augmented PID and feedforward inverse-model control	3.479	1.628	6.134	−0.227
8. Augmented PID and feedback inverse-model control	2.735	0	5.571	0

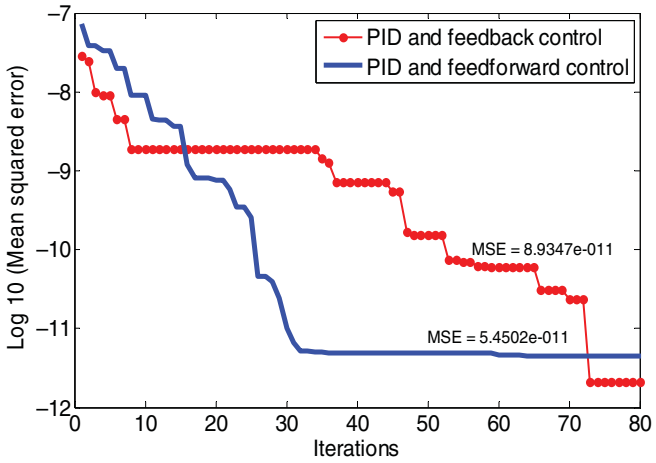


Fig. 18. Convergence graph of PSO-SF inverse-model-based control scheme.

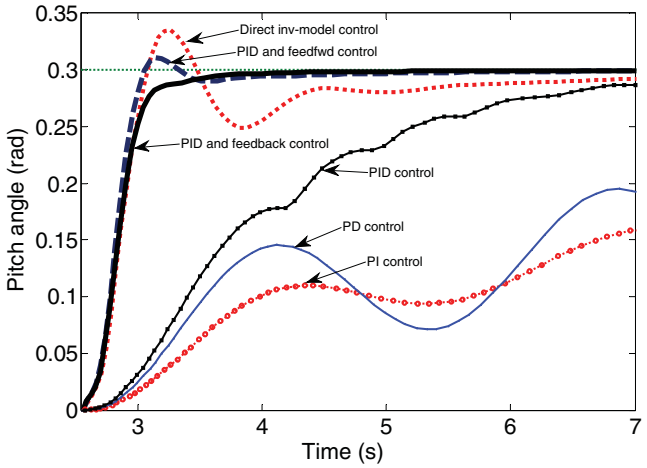


Fig. 19. System response with the various controllers.

the augmented PID and feedforward inverse control have resulted in improved performance. The augmented PID and feedback inverse-model control scheme has achieved superior performance with no overshoot and faster settling time and has led to good set-point tracking. The convergence graph of PSO-SF inverse model control scheme is presented in Fig. 18. It shows that the augmented PID and feedback inverse-model control has successfully achieved a good tracking response with smaller MSE value as compared with augmented PID and feedforward control.

7. Comparative Assessment

This section presents performance comparisons between classical PI, PD and PID control and inverse-model-based control schemes as shown in Fig. 19. Table VII summarises the quantitative results of the employed control strategies in terms of rise time, percentage overshoot, settling time and steady-state error of the system response. It is noted that the open-loop system response was unsatisfactory and was characterised by a significant overshoot 97.349% and long settling time. Results also showed that the open-loop performance can be improved by using classical control, PI control, PD control and PID control. However, as observed in Table VII and Fig. 19, the system response with these control schemes had longer rise times and longer settling times. The inverse-model-based control system has shown good performance in terms of reducing the oscillatory movement of the system. It is evident from the time-domain performance parameters that the augmented strategies employing inverse model control with the aid of PID control have resulted in better performance than using classical control alone. It is shown that the augmented dynamic model inversion was able to cancel the flexural effects of the controlled plant by constructing its inverse mapping. The augmented PID and inverse-model-based feedback control has shown superior results in output response tracking with no overshoot, smaller rise time (1.735 s), reduced settling time (5.571 s) and no steady-state error.

In order to investigate the robustness of the proposed inverse-model-based control scheme and its ability in disturbance rejection, impulsive disturbance signals of different amplitudes were applied to the system at 20 s, and the system performance was monitored. Four different levels of disturbance were used as shown in Table VIII. The system response thus monitored is shown in Fig. 20. It is noted that the system was able to reject the disturbance and recover within short time and achieve zero steady-state error. The system response settled within 7 s for the disturbance signal

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

Table VIII. Performance of the augmented PID and feedback inverse-model control scheme with applied disturbances.

Impulse input disturbance (volt)	Maximum pitch angle response (rad)	(%)	Recovery time (s)
30% × 0.3 = 0.09	0.3061	2.03	21.78
50% × 0.3 = 0.15	0.3099	3.30	22.72
80% × 0.3 = 0.24	0.3158	5.27	23.19
100% × 0.3 = 0.30	0.3375	12.50	27.36

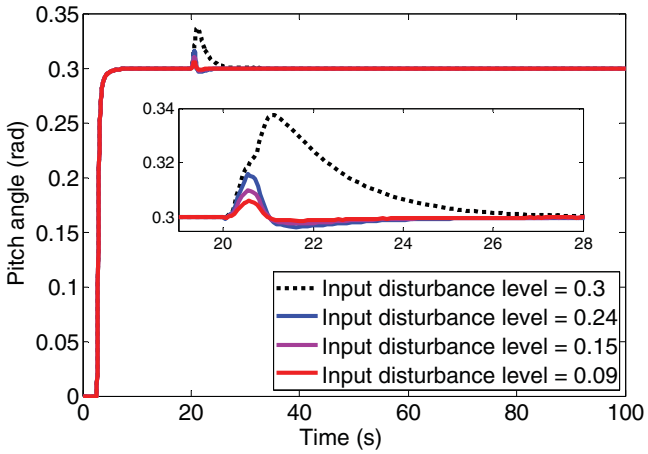


Fig. 20. Robust control of augmented PID and feedback inverse-model control scheme.

level of 0.3, which was 100% of the reference signal level. The settling times for the disturbance signal levels lower than 0.3 were significantly lower following application of the disturbance.

8. Conclusion

The PSO-SF algorithm has been successfully adopted as an intelligent technique for obtaining optimised PID parameters as well as inverse model of a twin rotor system. The classical control approach using PID control has not achieved satisfactory results in controlling the TRMS; it has resulted in longer response rise time and settling time. These problems have been effectively resolved by using augmented PID with feedforward and feedback inverse-model control. The latter approach has shown shorter settling time and rise time with smooth response. The results have clearly shown that better performance can be achieved with an intelligent inverse-model-based control approach as compared with a classical control approach for the TRMS. Moreover, the proposed inverse model control has shown robustness and disturbance rejection capability with impulsive disturbances.

References

1. Callender, D. R. Hartree and A. Porter, "Time-lag in a control system," *Phil. Trans. R. Soc.* **235**(756), 415–444 (1936).
2. Odwyer, *Handbook of PI and PID Controller Tuning Rules* (Imperial College Press, London, 2003).
3. C. Kuo, *Automatic Control Systems* (Prentice-Hall, New Jersey, 1995).
4. V. Kumar, K. P. S. Rana and V. Gupta, "Real-time performance evaluation of a Fuzzy PI + Fuzzy PD controller for liquid-level process," *Int. J. Intell. Control Syst.* **13**(2), 89–96 (2008).
5. J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME* **64**, 759–768 (1942).
6. G. H. Cohen and G. A. Coon, "Theoretical consideration of retarded control," *Trans. ASME* **75**, 827–834 (1953).
7. K. J. Åström and T. Hagglud, *PID Controller: Theory, Design and Tuning* (Research Triangle Park, NC, USA, 1995).
8. W. K. Ho, C. C. Hang and L. S. Cao, "Tuning of PID controllers based on gain and phase margins specifications," *Automatica* **31**(3), 497–502 (1995).
9. Z. L. Gaing, "A particle swarm optimisation approach for optimum design of PID controller in AVR system," *IEEE Tran. Energy Convers.* **19**, 384–391 (2004).
10. L. D. S. Coelho, "Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach," *Chaos Solitons Fractals* **39**(4), 1504–1514 (2009).
11. P. Van Overschee and B. De Moor, "RaPID: The End of Heuristic PID Tuning," *Preprints Proceeding PID '00: IFAC Workshop*, Terrassa, Spain (2000) pp. 687–692.
12. B. Ender, "Process control performance: Not as good as you think," *Control Eng.* **9**, 180–190 (1993).
13. T. Hägglund, "A control-loop performance monitor," *Control Eng. Pract.* **3**(11), 1543–1551 (1995).
14. M. Lin, S. Lakshminarayanan and G. P. Rangaiah, "A comparative study of recent/popular PID tuning rules for stable, first order plus dead time single input single output processes," *Ind. Eng. Chem. Res.* **47**(2), 344–368 (2008).
15. W.-D. Chang and S.-P. Shih, "PID controller design of nonlinear systems using an improved particle swarm optimization approach," *Commun. Nonlinear Sci. Numer. Simul.* **15**(11), 3632–3639 (2010).
16. A. Leva, S. Negro and A. Vittorio Papadopoulos, "PI/PID autotuning with contextual model parametrisation," *J. Process Control* **20**(4), 452–463 (2010).
17. C.-C. Kao, C.-W. Chuang and R.-F. Fung, "The self-tuning PID control in a slider-crank mechanism system by applying particle swarm optimization approach," *Mechatronics* **16**(8), 513–522 (2006).
18. T.-H. Kim, I. Maruta and T. Sugie, "Robust PID controller tuning based on the constrained particle swarm optimization," *Automatica* **44**(4), 1104–1110 (2008).
19. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, Perth, Australia (1995) pp. 1942–1948.
20. M. Clerc and J. Kennedy, "The particle swarm – explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evolutionary Comput.* **6**(1), 58–73 (2002).
21. R. C. Eberhart and Y. Shi, *Computational Intelligence Concepts to Implementations* (Morgan Kauffmann Publisher, USA, 2007).
22. S. F. Toha, I. Abd Latiff, M. Mohammad and M. O. Tokhi, "Parametric Modelling of a TRMS using Dynamic Spread Factor Particle Swarm Optimisation," *11th International Conference on Computer Modelling and Simulation (UKSIM09)*, Cambridge, UK (2009) pp. 95–100.
23. S. F. Toha and M. O. Tokhi, "Parametric modelling application to a twin rotor system using RLS, genetic and swarm optimisation techniques," *Proc. Inst. Mech. Eng.* **224**(9), 961–977 (2010), DOI: 10.1243/09544100JAERO09544706.
24. S. Zeng and J. Zhu, "Adaptive Compensated Dynamic Inversion Control for a Helicopter with

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press

Approximate Mathematical Model," *International Conference on Computational Intelligence for Modelling, Control and Automation,* Sydney, Australia (2006) pp. 208–208.
25. A. Rahideh, H. Shaheed and A. Bajodah, "Adaptive non-linear model inversion control of a twin rotor multi-input multi-output system using artificial intelligence," *Proc. Inst. Mech. Eng.* **221**(3), 343–351 (2007).
26. K. Danai, "Helicopter Rotor Tuning," **In**: *Vibration and Shock Handbook* (C. W. D. Silva, ed.) (Taylor & Francis, CRC Press, USA, 2005).
27. Feedback Instrument Limited, *Twin Rotor MIMO System Manual 33-007-0* (Sussex, UK, 1996).
28. S. Ahmad, A. Chipperfield and M. Tokhi, "Parametric modelling and dynamic characterization of a two-degree-of-freedom twin-rotor multi-input multi-output system," *Proc. Inst. Mech. Eng.* **215**(2), 63–78 (2001).
29. I. Abd Latiff and M. O. Tokhi, "Fast Convergence Strategy for Particle Swarm Optimization using Spreading Factor," *IEEE Congress on Evolutionary Computation*, Trondheim, Norway (2009) pp. 2693–2700.
30. M. R. Rapaic and Z. Kanovic, "Time-varying PSO – convergence analysis, convergence-related parameterization and new parameter adjustment schemes," *Inf. Process. Lett.* **109**(11), 548–552 (2009).
31. S. Lee, C. Ha and B. S. Kim, "Adaptive nonlinear control system design for helicopter robust command augmentation," *Aerosp. Sci. Technol.* **9**(3), 241–251 (2005).
32. L. Ljung, *System Identification: Theory for the User* (Prentice-Hall, Englewood Cliffs, New Jersey, 1999).

https://doi.org/10.1017/S0263574711000154 Published online by Cambridge University Press