

Hybrid spiral-bacterial foraging algorithm for a fuzzy control design of a flexible manipulator

Journal of Low Frequency Noise,
Vibration and Active Control
2021, Vol. 0(0) 1–19
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/14613484211035646
journals.sagepub.com/home/lfn


Ahmad Nor Kasruddin Nasir¹ , Mohd Ashraf Ahmad¹ and M.Osman Tokhi²

Abstract

A novel hybrid strategy combining a spiral dynamic algorithm (SDA) and a bacterial foraging algorithm (BFA) is presented in this article. A spiral model is incorporated into the chemotaxis of the BFA algorithm to enhance the capability of exploration and exploitation phases of both SDA and BFA with the aim to improve the fitness accuracy for the SDA and the convergence speed as well as the fitness accuracy for BFA. The proposed algorithm is tested with the Congress on Evolutionary Computation 2013 (CEC2013) benchmark functions, and its performance in terms of accuracy is compared with its predecessor algorithms. Consequently, for solving a complex engineering problem, the proposed algorithm is employed to obtain and optimise the fuzzy logic control parameters for the hub angle tracking of a flexible manipulator system. Analysis of the performance test with the benchmark functions shows that the proposed algorithm outperforms its predecessor algorithms with significant improvements and has a competitive performance compared to other well-known algorithms. In the context of solving a real-world problem, it is shown that the proposed algorithm achieves a faster convergence speed and a more accurate solution. Moreover, the time-domain response of the hub angle shows that the controller optimised by the proposed algorithm tracks the desired system response very well.

Keywords

Bacterial foraging algorithm, flexible manipulator system, PD-like fuzzy logic control, spiral dynamic

Introduction

Flexible manipulator systems are commonly found in industrial applications and space robotics. Compared to their rigid body counterparts, flexible manipulators offer several advantages. These include being light-in-weight, having low inertia and energy consumption, high-speed operation, increased productivity, and safer to handle. However, due to the flexible dynamics, distributed parameters and nonlinear nature of the system, dynamic modelling, and controller design of a flexible manipulator are challenging tasks. In the past, numerous conventional control schemes for flexible manipulator systems have been reported.¹ However, the conventional approaches require an accurate mathematical model to represent the dynamic characteristics of the flexible system, which is hard to acquire. Moreover, with the presence of uncertainties, which is normal for a real system, some of the physical parameters of the system are unavailable, and several assumptions have to be made during the dynamic modelling process. A promising control scheme for solving a complex and highly nonlinear system with uncertainty can be implemented through an intelligent control approach. These include fuzzy logic control (FLC),^{2,3} neural network control^{4–6} and hybrid proportional-derivative FLC.^{7,8}

The main advantage of an intelligent-based control over a conventional-based control is that it does not require an accurate mathematical model of the system. Intelligent-based control can handle nonlinear and complex problems more

¹Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, Pekan, Malaysia

²Department of Electrical and Electronic Engineering, School of Engineering, London South Bank University, London, UK

Corresponding author:

Ahmad Nor Kasruddin Nasir Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, Pekan Campus, Pekan 26600, Pahang, Malaysia.

Email: kasruddin@ump.edu.my and anq_qass1@yahoo.com



efficiently. In recent trends, intelligent-based control is commonly applied to solve real-world problems with a certain degree of uncertainty. In the case of FLC, the control scheme can be designed based on human expert knowledge of the system under control. However, for a highly complex system, expert knowledge of the system might be limited. This leads to the application of a trial and error method for determining fuzzy linguistic rules, membership functions, and input–output gains, which is difficult to apply. Consequently, it results in poor control performance. As a solution to the problem, a metaheuristic algorithm is applied to automatically seek an optimal fuzzy structure and parameters for the best system performance. With the current advances in processing and computing facilities, researchers are more attracted to apply the algorithm to find an optimal solution to such a problem.

Many researchers have reported the successful application of metaheuristic algorithms in optimisation of FLC-based control. Anh and Ahn (2011)⁹ applied a modified GA to optimise an inverse nonlinear autoregressive network with exogenous inputs (NARX) fuzzy model to control a pneumatic artificial muscle robot arm. They compared the proposed algorithm with the original genetic algorithm (GA). The result showed that the original GA had an accuracy problem due to landing on local optima. Mandal et al. (2015)¹⁰ compared the performances of the Bacterial Foraging Algorithm (BFA) and GA in the optimisation of real-time fuzzy-feedforward control for an electrohydraulic system with discontinuous nonlinearities. It was found that BFA outperformed GA and the designed controller was more robust than the original controller. Banerjee et al. (2011)¹¹ conducted a study comparing the performances of BFA and particle swarm optimization (PSO) in optimising a FLC for level control of a surge tank. It was found that BFA outperformed the PSO by resulting in a lower cost function value and more accurate result. Roy et al. (2010)¹² compared the performances of various evolutionary algorithms in optimising a hybrid FLC for automatic control of thermal generating plants. The study concluded that modified chaotic ant swarm optimization and BFA outperformed six categories of other optimisation algorithms including GA and PSO, in terms of accuracy and controller performance. Both algorithms were found to be more robust than other methods where they gave more consistent results when tested with 200 runs.

The literature review in the aforementioned paragraph indicates that BFA is a promising optimisation algorithm and has a competitive performance in solving various real-world problems. It is a well-known biological-inspired algorithm, which was introduced in 2002.¹³ Many researchers have pointed out that the constant step size C in the chemotaxis phase of BFA limits the performance of the BFA. Employing the approach, the convergence speed and final accuracy of the BFA cannot be achieved at the same time or they are in contradiction with each other. Defining a large C for the bacteria will result in faster convergence speed but low accuracy. On the contrary, by using a small C , a more accurate solution will be obtained but at a slow convergence speed since the bacteria need more steps to reach the optimum location hence increasing the total computation time to complete the process.

Various adaptive and hybrid strategies combining BFA and other metaheuristic algorithms with the aim to improve the performance of the original algorithms have been proposed by researchers.¹⁴ A hybrid approach may add extra complexity to the structure of the original algorithms. However, combining the advantages and unique features of the predecessor algorithms can possibly result in faster convergence and higher accuracy. Mishra (2005)¹⁵ combined BFA and a conventional least square algorithm to estimate the harmonic components in a power system current waveform. Both the BFA and least square were used to estimate the phase and amplitude of the harmonic, respectively. Kim and Cho (2007)¹⁶ hybridised BFA and GA and applied the resulting algorithm to optimise the controller parameters for an automatic voltage regulator. A hybrid strategy combining artificial immune system (AIS) and BFA was proposed.¹⁷ The clonal selection scheme was incorporated into bacteria chemotaxis to vary the bacteria step size and the algorithm was successfully applied to optimise the controller parameters of a three-phase induction motor. Biswas et al. (2007)¹⁸ proposed a hybrid BFA and PSO, where the PSO-based mutation scheme was combined with bacteria chemotaxis. Panigrahi and Pandi (2008)¹⁹ incorporated the Nelder-mead algorithm into BFA to improve the exploration strategy of bacteria within the search area.

Apart from BFA, another promising metaheuristic algorithm that is gaining attention from researchers worldwide is Spiral Dynamic Algorithm (SDA). It is an optimisation algorithm in which the strategy has been developed based on natural spiral phenomena.²⁰ One of the advantages of SDA is that it has a relatively simple structure hence it needs less computation time to complete the whole cycle of the search operation. However, there is a possibility for the algorithm to get trapped in local optima especially in the case of a more complex function with a high number of dimensions. This might occur possibly due to insufficient exploration process, or unbalanced exploration and exploitation of the search space. Another possibility is due to its deterministic strategy that relies only on the spiral model. During the diversification phase, all the agents move deterministically toward the centre of a spiral form. It guides the agents to move in a more limited area rather than being distributed thoroughly within the search space. For a function that has an uneven surface area and multiple global optimum points, the strategy may cause the agents to be trapped in a local optimum solution.

Few works on improving SDA through adaptive and hybrid strategies have been reported. Table 1 shows a summary of the previously published works on hybrid type SDA-BFA and the proposed algorithm. Previously published works have

utilised certain BFA strategies and incorporated into SDA such that the accuracy performance of SDA is improved. The main body of the developed algorithms consists of the SDA features and a few BFA features. In Refs. 21–24, only a chemotaxis part of the BFA was adopted to improve the exploration mechanism of the SDA. It resulted in the algorithm converging at a slower speed particularly at the beginning of the search process. For a function with multiple global points, the search agents tend to get trapped in a local optimum. In Ref. 25, on the other hand, the elimination and dispersal parts were adopted while the chemotaxis and reproduction phases were excluded. Moreover, only a deterministic spiral approach was considered in the strategy. The approach resulted in a slower convergence speed and insufficient exploration. Hence, there is high a possibility that the agents will get trapped in local optima. On the contrary, in the proposed algorithm, a spiral equation of the SDA is included in the BFA to improve the accuracy and convergence speed of the BFA as well as to increase the accuracy of the SDA. The whole original structure of the BFA is maintained and to complement the drawback of the BFA strategy, a deterministic spiral model of SDA is used with the BFA chemotaxis phase before the implementation of the reproduction, elimination, and dispersal phases. It is expected that the combination of both random and deterministic methods will balance both exploration and exploitation of the search space.

This article is based on preliminary work of the authors,²⁶ published in a conference where a spiral equation adopted from the SDA is incorporated into the chemotaxis phase of the BFA. The algorithm was applied to acquire a parametric model of a flexible manipulator system. This article is different from Ref. 26 such that it presents a comprehensive study and analysis on the accuracy performance of the algorithm tested on state-of-the-art CEC2013 benchmark functions. A statistical analysis based on Friedman and Wilcoxon Sign Rank tests is performed and compared with SDA, BFA, hybrid spiral dynamic bacteria chemotaxis (HSDBC),^{23,24} and Pattern Search (PS)²⁷ algorithms. It also presents an application of the algorithm to optimise a nonlinear hybrid Proportional-Derivative-like Fuzzy Logic Controller (PD-like FLC) for an input tracking problem of a flexible manipulator system. The tracking performance result of the optimised controllers is analysed and compared with SDA, BFA, and HSDBC algorithms. The proposed algorithm is referred to as the hybrid spiral-bacterial foraging algorithm (HSBFA). The development of the proposed algorithm is motivated by the following issues:

1. SDA has fast computation time and convergence speed, but it is low in accuracy.
2. BFA has good accuracy, but it has a long computation time and slow convergence speed.
3. The convergence speed and accuracy are the most important criteria for an optimization algorithm in solving any real-engineering problem.

Table 1. Review of existing hybrid BFA-SDA methods.

Reference	Essential features	Performances
HSDBF ²¹	To strengthen the exploration capability of the SDA algorithm. Spiral-based tumble and swim strategies for BFA were adopted prior to the execution of SDA.	The algorithm got trapped into local optima. Increased total computation time of SDA. Inconsistent results due to the adoption of a spiral model.
HBCSD ²²	To improve the exploration strategy of SDA. Random-based tumble and swim strategies of BFA were adopted prior to the execution of SDA.	Increased accuracy of SDA. The algorithm got trapped into local optima. Increased total computation time of SDA.
HSDBC ^{23,24}	To enhance the exploration strategy of SDA. Spiral-based and random-based swim strategies of BFA incorporated into SDA.	Improved the accuracy of SDA. The algorithm got trapped into the local optima. Increased total computation time of SDA.
ISDA ²⁵	To improve the performance of SDA. Elimination and dispersal strategies of BFA incorporated into SDA to solve premature convergence.	Increased the accuracy of SDA. The algorithm had a slower convergence speed than SDA. Increased total computation time of SDA.
Proposed algorithm (HSBFA)	To improve the accuracy of BFA and SDA. To expedite the convergence speed of BFA. The spiral equation of SDA adopted prior to the implementation of reproduction, elimination and dispersal strategies of BFA.	Increased accuracy in finding global optima solutions compared to both SDA and BFA. Expedited the convergence speed of BFA. Increased total computation time of BFA. Increased number of user-define parameters of BFA.

BFA: bacterial foraging algorithm, SDA: spiral dynamic algorithm; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

The aim of this article is first to propose an improved version of a hybrid SDA-BFA algorithm which can improve the accuracy performance of SDA and BFA algorithms and to improve the convergence speed of BFA. The proposed algorithm is then tested with CEC2013 benchmark functions to show its effectiveness in obtaining near-global optimal solutions in comparison to SDA, BFA, HSDBC, and PS algorithms. The second aim is to show its effectiveness in a PD-like FLC control design of a flexible manipulator system. Considering no expert knowledge is present, the approach allows automatic and optimal attainment of linguistic rules for the FLC and their corresponding fuzzy weight parameters, and fuzzy input-output gains for the best tracking performance. The rest of the article is organised as follows. The proposed HSBFA is described in the Hybrid Spiral-Bacterial Foraging Algorithm section while the analysis on its accuracy performance tested on CEC2013 benchmark functions is presented in the Performance Test on CEC2013 Benchmark Functions section. The application of the proposed HSBFA algorithm to optimise the hybrid PD-like FLC for the flexible manipulator system is presented in the PD-Like FLC Optimisation for Flexible Manipulator section. Finally, the conclusions drawn from the work are presented in the Conclusion section.

Hybrid spiral-bacterial foraging algorithm

The proposed hybrid spiral-bacterial foraging algorithm (HSBFA) is developed based on the original SDA and BFA. A detailed explanation of the proposed HSBFA is presented in this section. In the proposed HSBFA, the three essential phases such as chemotaxis, reproduction, and elimination-dispersal of BFA are retained. The only change is that the spiral model of SDA is incorporated into the chemotaxis phase to improve the position of bacteria. The main objective of combining the spiral feature into the BFA structure is to have a balance between exploration and exploitation phases such that faster convergence and better accuracy are achieved. This modification can help the BFA in several aspects, and these can be described as follows.

First, the original strategy of BFA is based on the random movement of individual bacterium around its local position that happens in the chemotaxis phase. However, the interaction of the best bacterium and other bacteria in that particular iteration does not occur. Some of the bacteria tend to search in locations away from the global optimum location and may result in slow convergence. In SDA, the spiral centre is always considered as a point with the best fitness in the current iteration. Since the motion of the search point is towards the spiral centre, in general, the movement of search points is always guided by the best optimum location in that particular iteration. Incorporating the spiral model can guide the bacteria movement towards the global best bacterium position in the current iteration. This lets all other bacteria search possible optimum location nearer to the global best solution and consequently, it speeds up the algorithm's convergence.

Second, employing a spiral model, bacteria are guided to move in a spiral way, and the spiral trajectory is determined by a user-defined spiral angle and radius. Therefore, instead of using the solely random approach in determining the bacteria motion, by employing the spiral model, the bacteria movement is enhanced through a combination of random and deterministic approaches. The deterministic approach always guides the bacteria towards the best bacterium while the stochastic approach moves the bacteria in a random direction.

Third, the exploration and exploitation of the BFA can be enhanced through diversification and intensification of the spiral approach. By employing the spiral strategy, the bacteria tend to move diversely throughout the search area and move intensely in a more confined space at the initial and final stages, respectively, while at the same time the random strategy is applied where it can move the bacteria freely throughout the search area during the whole search operation.

Fourth, in the original BFA strategy, the bacteria step size is constant where it prevents the simultaneous achievement of faster convergence and high accuracy. Applying the spiral model, a combination of constant step size and dynamic step size is incorporated. The bacteria move with a large step at the early stage of the search operation and the step size is reduced as the bacteria approach the final phase of the search operation. When the bacteria are far from the global optimum location, they move faster with a large step size but they move slower with a smaller step size when they approach the global optimum location. Hence, faster convergence and a highly accurate solution can be achieved even if the global optimum solution is located at a remote location. By having an adaptive step size in the algorithm, the oscillation problem near the optimum location can be resolved, and a more stable convergence can be achieved.

Parameters of the HSBFA are shown in [Table 2](#), and its corresponding flowchart is presented in [Figure 1](#).

In general, the fitness of each bacterium is obtained after the initialisation step to determine the best bacterium. A bacterium is considered the best bacterium if it has the highest fitness or the lowest value of cost function. Then, the best bacterium is defined as x^* , which is a centre point of the spiral motion. Unlike the standard BFA and other hybrid BFA-SDA variants, at this stage, HSBFA memorises the current best location of a bacterium, x^* , in the whole bacteria population. The chemotaxis phase consists of the tumble and swims actions. Before that, the algorithm saves the location and fitness value of each bacterium in the population. In the tumble, each bacterium randomly moves linearly one step ahead around its current

Table 2. Parameters of the hybrid spiral-bacterial foraging algorithm.

Parameter	Description
S	Total number of bacteria in the population
p	Dimension of the search space
C	Bacteria step size
P_{ed}	Elimination-dispersal probability
N_S	Swimming length
N_c	Number of chemotactic steps
N_{re}	Number of reproduction
N_{ed}	Number of elimination-dispersal
θ_s	Spiral angle or rotation angle
r_s	Spiral radius

location to search for another better solution. The fitness value at a new location after the tumble for each bacterium is saved before a swim action is executed.

In the swim action, the bacterium will move linearly several steps ahead in the same direction, which is similar to tumble action, if it has a better fitness value than the previous location. The bacterium will swim until it reaches the maximum value of swim Q_s , defined at the initialization stage. The bacterium movement for the tumble and swim actions is based on the constant step size, C , defined at the initialization stage. After completing the tumble and swim actions, each bacterium is spirally directed towards the global best position by utilizing a spiral equation adopted from the SDA algorithm as highlighted in the flowchart. The position update equation generating the spiral trajectory is presented as equation (1)

$$\theta^i(j+1, k, l) = S_n(r_s, \theta_s) \theta^i(j+1, k, l) - (S_n(r_s, \theta_s) - I_n) x^* \quad (1)$$

where $\theta^i(j+1, k, l)$ is the location of the i th bacterium in j th chemotaxis, k th reproduction, and l th elimination-dispersal phases. $S_n(r_s, \theta_s)$ is the rotational matrix defined in terms of spiral radius and spiral angle. I_n is the identity $n \times n$ matrix, where n is the number of dimensions.

This is to guide the movement of the whole bacteria population based on the best bacterium, x^* , which finally brings the bacteria to the global best location in the search space. Apart from that, by applying the spiral equation, the bacteria step size will be varied throughout the search process. The distance between the two points is larger at the outer layers and it is closer at the inner layers of the spiral form. This strategy complements the drawback of using constant bacteria step size in the standard BFA. Each bacterium moves with a larger step size at the early stage of the search operation and moves with a smaller step size as the number of iterations increases. Thus, the bacteria can reach all possible solutions at any remote location within the search space. More importantly, by implementing this approach, a balanced exploration and exploitation strategy can be achieved and it will result in better convergence speed with higher accuracy.

Next, the fitnesses of each bacterium are evaluated again to determine the best bacterium in the current iteration before starting a new reproduction phase. The best bacterium is assigned as x^* and will be used in the next chemotaxis phase. All these steps are continuously repeated until the algorithm is terminated. By using the proposed approach, the bacteria are guided based on the local best position found by each bacterium through the random approach of the chemotactic strategy as well as the global best position in the spiral dynamic model. Moreover, the step size of the bacteria is varied based on their distance from the global optima location through the spiral model.

Performance test on CEC2013 benchmark functions

In this section, a set of benchmark functions with 10 and 30 dimensions are used to assess the performance of the proposed algorithm in finding a near-global optimum solution. A performance comparison between the proposed algorithm and the original SDA, BFA, HSDBC,²⁴ and PS algorithms²⁷ is conducted. In particular, a set of benchmark functions from the 2013 IEEE Congress on Evolutionary Computation²⁸ is used. These benchmark functions have 28 different functions with various properties, landscapes, and complex features.

In order to get a fair comparison, the parameters of the proposed hybrid algorithm and the original SDA, BFA, HSDBC, and PS algorithms are chosen after conducting exhaustive trial and error experiments such that the best performance is obtained. The initial BFA parameters were set based on the value proposed in Refs. 13 and 14 while for HSDBC and SDA, they were set

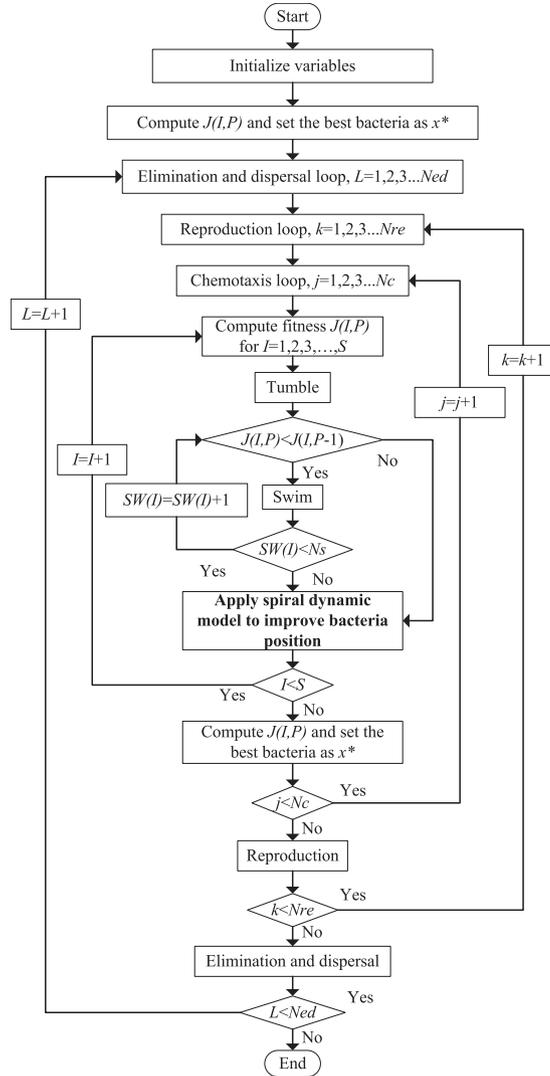


Figure 1. Flowchart of the proposed hybrid spiral-bacterial foraging algorithm.

based on the best value from previously conducted experiments.²⁴ The number of bacteria swims, chemotactic, reproduction, elimination, and dispersal for BFA and HSBFA and the number of iterations for HSDBC and SDA were set just enough to meet the maximum number of evaluations as required for the CEC2013 benchmark functions test. The parameter value of the bacteria step length for BFA, spiral radius, and spiral angle for SDA, HSDBC, and HSBFA were then stochastically fine-tuned to meet the optimal performance. The parameters for HSDBC were the same as SDA except for the two additional parameters defined for a chemotactic strategy, which were initially set based on Refs. 13 and 14 before they were fine-tuned.

The user-defined parameters are depicted in Table 3. For a fair performance comparison, in the case of the 10 dimensions, the number of bacteria S and the maximum number of fitness evaluations (FEs) were set to 50 and 100,000, respectively, for all algorithms. Meanwhile, in the case of 30 dimensions, the number of bacteria S and the maximum number of FEs were set to 100 and 300,000, respectively, for all the algorithms. The objective function was set to zero if it reaches a value less than 10^{-8} for both cases of 10 and 30 dimensions. For simplicity, the number of search points in SDA was treated as the number of bacteria S . The simulation was run for 25 trials and the mean value of the objective function was recorded. The non-parametric Friedman test was used to evaluate the performance and the ranking of the algorithms in terms of their capability to find the global optimal solution.

Table 4 shows the statistical analysis of the algorithms after performing the CEC2013 benchmark functions with 10 and 30 dimensions. The table presents the mean value and standard deviation (in bracket) of the final accuracy based on 25 runs. The mean value depicts the accuracy of the final solution. The best mean value between all four algorithms is highlighted in

Table 3. User-defined parameters of the algorithms.

SDA	BFA	HSBFA	HSDBC
$r_s = 0.96,$ $\theta_s = \pi$	$C = 0.5, P_{ed} = 0.25, N_s=4, N_c=50,$ $N_{re}=3, N_{ed}=6$	$C = 1 \times 10^{-6}, N_s=4, N_c=50, N_{re}=4, N_{ed}=6$ $r_s = 0.94, \theta_s = \pi/2$	$C = 1, N_s=1, r_s = 0.96,$ $\theta_s = \pi,$

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

bold font. The statistical result of the PS algorithm tested on CEC2013 as shown in the table is adopted from the work published in Ref. 29. It is noted that, in the case of 10 dimensions, all algorithms except for BFA shared the best mean for the function f_1 . SDA achieved the best mean for function f_{28} while BFA achieved the best mean for functions f_{22} and f_{23} . SDA and HSDBC shared the best mean for function f_{25} . HSDBC achieved the best mean for functions f_3 and f_{27} . The performance of PS was the second best among the contested algorithms where it achieved the best for functions $f_5, f_8, f_{11}, f_{14}, f_{17}, f_{19}, f_{21}, f_{24},$ and f_{26} . HSBFA achieved the best mean for the rest of the functions. Meanwhile, in the case of 30 dimensions, the SDA, HSDBC, and PS shared the best mean for function f_1 . The SDA achieved the lowest mean for functions $f_{18}, f_{20},$ and f_{21} while BFA achieved the lowest mean for functions f_{15} and f_{23} . The performance of HSBFA was the second best among the compared algorithms where it achieved the best mean for functions $f_4, f_9, f_{13}, f_{16}, f_{24},$ and f_{26} . HSDBC achieved the best mean for functions f_{12} and f_{27} . Moreover, it shared the best mean with SDA for function f_{25} . PS achieved the best mean for the rest of the functions.

Table 5 depicts an analysis of the Friedman test conducted on the CEC2013 benchmark functions with 10 and 30 dimensions using the numerical results in Table 4. The Friedman test aims to show the rank of all the contested algorithms based on their ability to find a near-global optimum solution. The alpha value of 5% degree-of-significance is defined for all the tests, which gives a 95% confidence interval for the produced results. An algorithm is considered to have better performance and a significant difference compared to its opponents if the two-tailed, p -value is less than 0.05 with a 5% degree-of-significance. The best rank of the algorithm is stated in bold font. Note that, in both cases of 10 and 30 dimensions, HSBFA achieved the best rank followed by PS, HSDBC, SDA, and BFA. The two-tailed, p -value for all five cases was less than 0.05, which indicates that the difference was significant. The chi-square χ^2 indicates a statistical test value or a distribution of a sum of a square of random variables.

Table 6 depicts an analysis of the Wilcoxon Signed-Rank test evaluated on the CEC2013 benchmark functions for the 10 and 30 dimensions using the numerical results in Table 4. The test aims to show the best performance between the two algorithms. The R+ and R- indicate the sum of positive and negative ranks, respectively. Z indicates the standard deviation score of the test. In Table 6, HSBFA is compared with other contested algorithms. Notice that, for both of the 10- and 30-dimension benchmark functions, all the two-tailed, p -values were less than 0.05 if compared with SDA and BFA. Comparing HSBFA with HSDBC, the p -value was less than 0.05 only for the 10 dimensions. The p -value was greater than 0.05 for the 30 dimensions of HSDBC and both 10 and 30 dimensions of PS. This reveals that HSBFA has a significant improvement in terms of finding a near-global optimum point for both the 10 and 30 dimensions compared to SDA and BFA. The HSBFA had significant improvement only for 10 dimensions compared to HSDBC. On the contrary, the result with HSBFA was not significant compared to the results of 10 and 30 dimensions of PS and 30 dimensions of HSDBC.

PD-like FLC optimisation for flexible manipulator

In this study, a composite control scheme namely Proportional-Derivative-like Fuzzy Logic Control (PD-like FLC) is used for position tracking of a hub angle. Figure 2 shows the hub angle feedback control block diagram of a PD-like FLC.³⁰

The block diagram comprises five components and their explanation is presented as in Table 7. A desired bang-bang type signal is defined as an input to the flexible manipulator system (FMS) or flexible robotic manipulator (FRM). It is considered as a desired hub angle trajectory of the FMS. Based on the input, a dc motor actuator of the system drives the flexible link of the FMS. In response to the movement, an encoder and a tachometer that are attached to the system measure hub angle and hub velocity responses, respectively. The response of the hub angle is compared with the desired bang-bang input and it gives an error reading or a difference between the actual and the desired hub angle responses. The error and the hub velocity response are then applied into the PD-like FLC which acts as a compensator. The metaheuristic algorithms which represent the proposed HSBFA and other algorithms under study get into the cycle. They receive the error reading of hub angle as an input while optimising the compensator parameters as stated in Table 7. Finally, the output signal of the compensator is applied into the FMS and it completes one cycle of the feedback control system.

Table 4. Statistical result for 10 and 30 dimensions f_1 to f_{28} .

D	F	SDA	BFA	HSBFA	HSDBC	PS
D10	f_1	0.00E + 00	9.57E + 03	0.00E + 00	0.00E + 00	0.00E + 00
	f_2	2.30E + 06	1.70E + 07	1.05E + 05	2.67E + 05	1.71E + 05
	f_3	4.42E + 09	2.83E + 10	8.72E + 07	5.54E + 07	1.30E + 09
	f_4	1.06E + 05	2.20E + 04	2.92E - 03	1.65E + 03	4.11E + 04
	f_5	3.91E + 01	1.70E + 03	2.56E - 05	3.08E + 01	0.00E + 00
	f_6	1.23E + 01	6.52E + 02	4.77E + 00	2.72E + 01	1.00E + 01
	f_7	1.33E + 02	6.27E + 03	9.07E + 01	1.53E + 02	1.09E + 02
	f_8	2.12E + 01	2.08E + 01	2.07E + 01	2.09E + 01	2.06E + 01
	f_9	6.62E + 00	1.04E + 01	5.04E + 00	6.46E + 00	8.30E + 00
	f_{10}	1.18E + 00	8.97E + 02	2.43E - 01	8.46E - 01	1.90E + 00
	f_{11}	3.67E + 01	1.70E + 02	4.62E + 01	3.87E + 01	1.88E + 01
	f_{12}	3.76E + 01	1.55E + 02	3.13E + 01	4.25E + 01	5.17E + 01
	f_{13}	5.67E + 01	2.17E + 02	5.45E + 01	6.32E + 01	9.05E + 01
	f_{14}	1.10E + 03	9.57E + 02	9.78E + 02	1.30E + 03	5.44E + 02
	f_{15}	1.16E + 03	9.08E + 02	8.51E + 02	1.13E + 03	1.41E + 03
	f_{16}	4.43E - 01	4.54E - 01	1.82E - 01	6.58E - 01	1.00E + 00
	f_{17}	4.62E + 01	3.75E + 02	3.46E + 01	7.04E + 01	2.49E + 01
	f_{18}	3.58E + 01	3.71E + 02	3.39E + 01	7.89E + 01	7.47E + 01
	f_{19}	1.84E + 00	1.48E + 04	1.81E + 00	1.95E + 00	1.20E + 00
	f_{20}	3.96E + 00	4.98E + 00	3.92E + 00	4.07E + 00	4.80E + 00
	f_{21}	3.96E + 02	8.37E + 02	3.92E + 02	3.96E + 02	3.63E + 02
	f_{22}	1.52E + 03	1.21E + 03	1.22E + 03	1.38E + 03	6.33E + 03
	f_{23}	1.36E + 03	1.20E + 03	1.34E + 03	1.37E + 03	1.83E + 03
	f_{24}	2.17E + 02	2.44E + 02	2.15E + 02	2.15E + 02	2.00E + 02
	f_{25}	2.15E + 02	2.37E + 02	2.16E + 02	2.15E + 02	2.22E + 02
	f_{26}	2.41E + 02	2.64E + 02	2.32E + 02	2.38E + 02	1.97E + 02
	f_{27}	4.50E + 02	8.51E + 02	4.49E + 02	4.42E + 02	5.08E + 02
	f_{28}	5.13E + 02	1.35E + 03	5.28E + 02	5.80E + 02	6.49E + 02
D30	f_1	0.00E + 00	7.46E + 04	1.14E - 03	0.00E + 00	0.00E + 00
	f_2	3.24E + 07	9.13E + 08	1.46E + 06	2.05E + 06	1.10E + 06
	f_3	1.01E + 11	9.92E + 16	7.34E + 09	4.38E + 09	1.70E + 09
	f_4	1.50E + 05	1.08E + 05	1.18E + 00	1.86E + 03	1.31E + 05
	f_5	1.82E + 02	2.34E + 04	2.26E - 02	1.78E + 02	0.00E + 00
	f_6	4.33E + 01	1.33E + 04	5.75E + 01	5.48E + 01	1.00E + 01
	f_7	3.48E + 02	3.92E + 06	2.93E + 02	3.15E + 02	1.48E + 02
	f_8	2.16E + 01	2.12E + 01	2.13E + 01	2.14E + 01	2.11E + 01
	f_9	3.01E + 01	4.09E + 01	2.90E + 01	3.03E + 01	3.17E + 01
	f_{10}	3.92E + 01	9.69E + 03	1.29E + 00	1.93E + 01	1.00E - 01
	f_{11}	2.81E + 02	1.08E + 03	2.77E + 02	2.69E + 02	5.38E + 01
	f_{12}	2.81E + 02	1.06E + 03	2.72E + 02	2.61E + 02	2.91E + 02
	f_{13}	4.09E + 02	1.26E + 03	3.67E + 02	4.02E + 02	4.13E + 02
	f_{14}	4.57E + 03	3.73E + 03	4.29E + 03	4.51E + 03	1.56E + 03
	f_{15}	4.60E + 03	3.81E + 03	3.92E + 03	4.59E + 03	4.69E + 03
	f_{16}	1.04E + 00	1.08E + 00	4.85E - 01	7.06E - 01	1.00E + 00
	f_{17}	2.92E + 02	2.43E + 03	3.22E + 02	4.04E + 02	8.31E + 01
	f_{18}	2.91E + 02	2.36E + 03	3.04E + 02	4.61E + 02	3.53E + 02
	f_{19}	2.03E + 01	2.54E + 06	1.47E + 01	2.46E + 01	3.60E + 00
	f_{20}	1.46E + 01	1.50E + 01	1.47E + 01	1.48E + 01	1.50E + 01
	f_{21}	3.21E + 02	4.68E + 03	3.59E + 02	3.62E + 02	3.89E + 02
	f_{22}	5.12E + 03	4.68E + 03	4.98E + 03	4.90E + 03	1.54E + 03
	f_{23}	4.90E + 03	4.81E + 03	4.85E + 03	4.86E + 03	6.12E + 03
	f_{24}	2.76E + 02	4.67E + 02	2.75E + 02	2.76E + 02	3.02E + 02
	f_{25}	2.74E + 02	4.26E + 02	2.76E + 02	2.74E + 02	3.16E + 02
	f_{26}	3.60E + 02	3.88E + 02	3.26E + 02	3.49E + 02	3.70E + 02
	f_{27}	1.05E + 03	1.70E + 03	1.07E + 03	1.02E + 03	1.17E + 03
	f_{28}	1.56E + 03	8.03E + 03	1.26E + 03	9.09E + 02	8.73E + 02

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

Table 5. Analysis of the Friedman test for the CEC2013 functions.

Algorithm	10 dimensions		30 dimensions	
	$p < 0.05$	$\chi^2 < 28.75$	$p < 0.05$	$\chi^2 < 38.54$
	Mean rank		Mean rank	
HSBFA	1.77		2.29	
PS	2.70		2.55	
HSDBC	3.09		2.68	
SDA	3.13		3.18	
BFA	4.32		4.30	

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

Table 6. Analysis of the Wilcoxon signed-rank test for CEC2013 functions.

Algorithm	Dim	Wilcoxon signed-rank test			
		R+	R-	p	Z
SDA-HSBFA	10	339	39	$p < 0.05$	-3.603
	30	329 Significant	77	$p < 0.05$	-2.869
BFA-HSBFA	10	382	24	$p < 0.05$	-4.076
	30	372 Significant	34	$p < 0.05$	-3.848
HSDBC-HSBFA	10	321	57	$p < 0.05$	-3.171
	30	276 Non-significant	130	$p = 0.09$	-1.662
PS-HSBFA	10	261	117	$p = 0.08$	-1.729
	30	189 Non-significant	217	$p = 0.75$	-0.318

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

Flexible manipulator system

Figure 3 shows a schematic representation of the single-link FMS. The dimensions of the beam such as the width, the length, and the thickness, are defined as 19.008 mm, 900 mm, and 3.2004 mm, respectively. L is the cross-sectional area of the beam. The area moment of inertia $I = 5.19 \times 10^{-11} \text{ m}^4$, mass density per unit volume, $\rho = 2710 \text{ kgm}^{-3}$, the Young modulus, $E = 71 \times 10^9 \text{ Nm}^{-2}$, and the hub inertia $I_h = 5.86 \times 10^{-4} \text{ kgm}^2$ are further specifications of the system. $w(x,t)$ denotes elastic deflection of a point on the beam at a distance x from the hub at a time t , and m_p denotes the mass of a payload. When a torque, τ is applied at the hub, an actuator drives the flexible beam to rotate at an angle, θ_t at a specific time, t . The rotating axis of the beam changes accordingly from the original axis, (X_0, Y_0) , to the new axis, (X, Y) .

Figure 4 shows a detailed description of the flexible manipulator system.³¹ The main component of the system is a flexible link that is made of aluminium rods. It is planarly constrained with flexible dynamics in the horizontal plane and stiff in vertical a motion. An electric motor, which acted as an actuator to the system, is attached to the hub of the flexible link. A tachometer is attached to the electric motor shaft to measure the hub angular velocity. The output of the tachometer is then amplified by a linear amplifier and is applied to a low pass filter, LPF2 to attenuate noise from the amplified signal. An encoder sensor with a resolution of 2048 pulses-per-one-revolution is attached to the electric motor

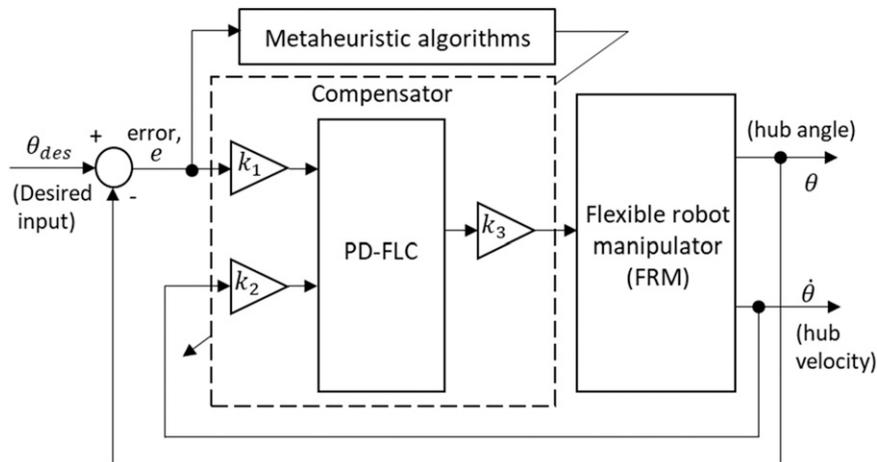


Figure 2. Block diagram of the hub angle feedback control.

Table 7. Components of the hub angle feedback control.

No	Component	Explanation
1	Input	The desired input is a bang-bang type input as shown in Figures 7(a) and (b).
2	Flexible manipulator system/flexible robot manipulator (FMS)	FMS is the system under control. The hub angle is considered as the output of interest. It should mimic the desired bang-bang input. Schematic representation and diagram of the single-link FMS are shown in Figures 3 and 4 respectively.
3	Proportional-derivative (PD) controller	Consists of a proportional gain, k_1 and a derivative gain, k_2 . Both gains are tuneable constants.
4	Fuzzy logic controller (FLC)	Comprises of fuzzifier, fuzzy rule base, inference engine and defuzzifier. The FLC comprises of 25 fuzzy rules. Each fuzzy rule has 3 tuneable fuzzy sets and 1 tuneable firing weight. Therefore, there are 100 tuneable constants for the FLC.
5	Metaheuristic algorithm (the proposed HSBFA)	It should optimise 1 proportional gain, 1 derivative gain, 75 fuzzy sets, 25 firing weights and 1 fuzzy output gain k_3 . Therefore, the HSBFA has to optimise 103 constants. An optimal combination of all parameters value will result a good hub angle performance.

FLC: fuzzy logic control; HSBFA: hybrid spiral-bacterial foraging algorithm.

shaft to measure the hub position. The output voltage of the encoder is converted from digital to analogue (D/A) signal by a digital voltage processor or (D/A) converter. It is then applied to another low pass filter, LPF3 for removing noise from the encoder output signal. In order to force the electric motor to rotate in both clockwise and counter-clockwise directions, and enable the link to swing and bend, a bi-directional drive amplifier mounted on a printed circuit board was used. The vibration of the link is measured by an accelerometer sensor attached at the end of the flexible link. The accelerometer consists of a built-in Field Effect Transistor (FET) source follower. It lowers down the impedance level at the output and this avoids the signal distortion. The output signal of the accelerometer is amplified by a linear amplifier and applied into a bandpass filter, BPF to measure the vibration of the flexible link. A personal computer (PC) with a 500 MHz processor is used for the implementation of the control algorithm. An interfacing unit PCL818G card is used to communicate between the PC and the flexible manipulator system and for conversion signals from analogue to digital and vice versa. The output signal of the interfacing unit is applied to a motor driver amplifier in a form of a voltage signal. The motor driver produces a current signal proportional to the input voltage and amplifies the signal before being applied to the electric motor.³² The signal is also fed back through another low pass filter, LPF1 into the interfacing unit for the purpose of monitoring the applied current signal into the electric motor. Installed in the PC is Matlab software, which served as a platform for controller design.

In the perspective of a control theory, the important control variables for a flexible manipulator system are the hub-angle and end-point acceleration or vibration of the flexible link. In a real-world application, a control scheme is required to deal with the hub angle position tracking problem while reducing the vibration of the flexible link.^{33,34}

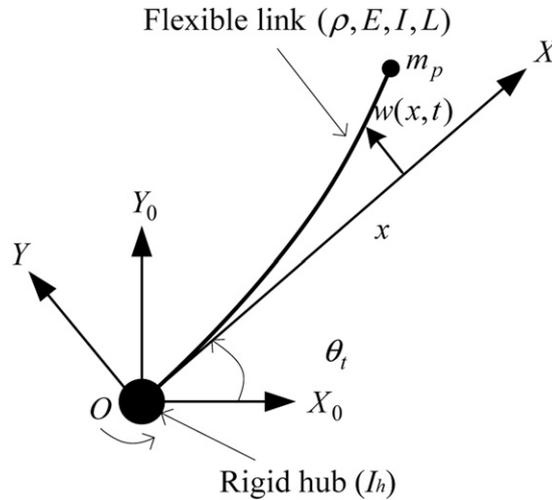


Figure 3. Schematic representation of a single-link flexible manipulator system.

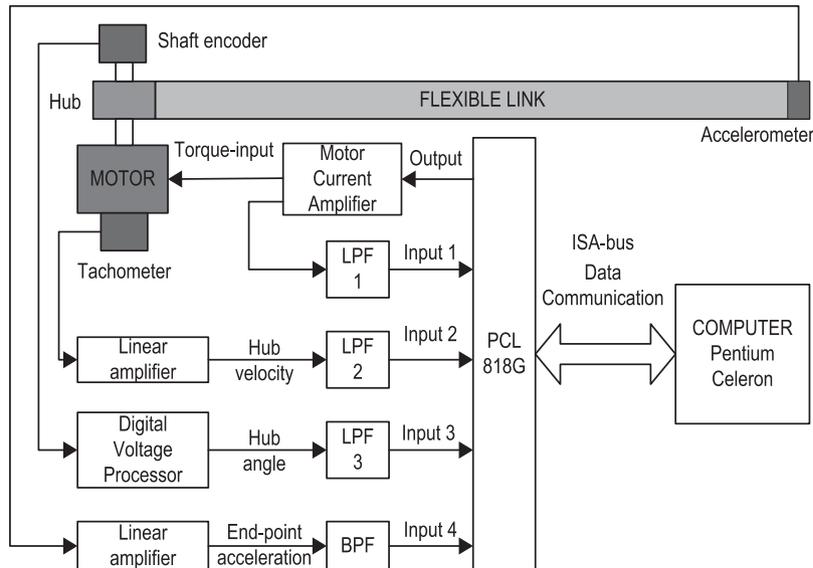


Figure 4. Schematic diagram of the single-link flexible manipulator.³¹

Proportional-derivative control

The parameters k_1 and k_2 correspond to the proportional and derivative gains, respectively. Here, the proportional gain k_1 is multiplied with the hub angle error and the derivative gain k_2 is multiplied with the hub angle velocity that are fed back directly from the system. Derivative of a hub angle represents a hub angle velocity. Since the control scheme is a proportional-derivative (PD)-fuzzy type, the feedback signals from the hub angle and hub velocity are considered as the inputs to the fuzzy system. In other words, both k_1 and k_2 can be represented as scaling factors for the fuzzy controller inputs. These scaling factors are important in deciding the optimal boundary of actual physical values that are mapped into a normalised universe of discourse. Moreover, parameter k_3 is a scaling factor of the FLC output. It projects the normalised output value of FLC into its actual physical value. This physical value corresponds to the input torque of the flexible manipulator system. The PD control equation, $G_{PD}(t)$ is shown as equation (2)

$$G_{PD}(t) = e(t)k_1 + \frac{de(t)}{dt}k_2 \tag{2}$$

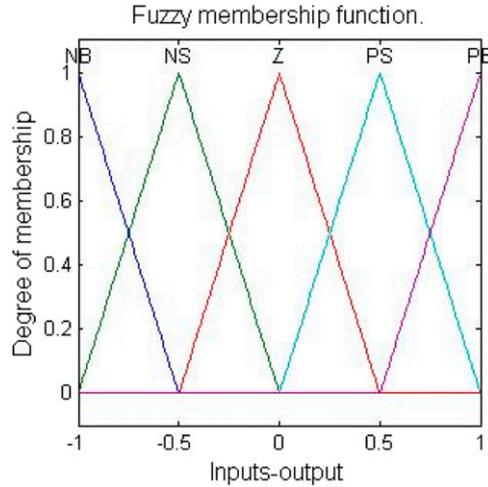


Figure 5. Membership functions for the inputs and output of the fuzzy logic controller.

Table 8. Parameter values for SDA, BFA, HSBFA, and HSDBC.

SDA	BFA	HSBFA	HSDBC
$S = 50, r_s = 0.95\theta_s = \pi/4, iter_{tot}=1000$	$S = 50, C = 0.5, P_{ed} = 0.25, N_s=4, N_c=50, N_{re}=3, N_{ed}=6$	$S = 50, C = 1 \times 10^{-6}N_s=4, N_c=50, N_{re}=4, N_{ed}=6r_s = 0.94, \theta_s = \pi/2$	$S = 50, N_s=4, r_s = 0.95, \theta_s = \pi/4C=8 \times 10^{-3}$

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

Table 9. Optimised gain values for PD-like FLC.

Parameter	SDA	BFA	HSBFA	HSDBC
k_1	0.0297	0.0164	0.0362	0.0230
k_2	0.0025	0.0023	0.0029	0.0018
k_3	0.2800	0.3247	0.3696	0.4452

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm; FLC: fuzzy logic control; HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

Fuzzy logic control

Membership function. Considering the performance of the fuzzy logic control (FLC) and its processing time, five triangular type membership functions are employed with a 0.5 cross-point and uniformly distributed throughout the universe of discourse within a range of $[-1, 1]$ as presented in Figure 5. The set of symbols $\{NH, NL, ZE, PL, PH\}$ represent the fuzzy sets {negative high, negative low, zero, positive low, positive high}, respectively. The same membership function type is used for fuzzy output and both of the fuzzy inputs. As a result, 25 fuzzy rules are generated in this FLC.

Linguistic fuzzy rule. In developing the FLC rules, the linguistic rules of IF-THEN statements are used to construct a relationship between FLC inputs and output. Let the symbols $\varepsilon, v,$ and τ represent the hub angle error, hub velocity, and input torque of the flexible manipulator, respectively. Then, an example of a linguistic rule is given as equation (3)

$$\text{IF } \varepsilon \text{ is } P \text{ AND } v \text{ is } Q \text{ THEN } \tau \text{ is } R \quad (3)$$

where $P, Q,$ and R are the linguistic variables denoted by the fuzzy sets $\{NH, NL, ZE, PL, PH\}$. Here, the Mamdani inference method is used, which is based on the *min*-type operator for the implication and the *max*-type operator for the aggregation. Finally, a centroid defuzzification method is chosen to transform the inference engine output based on a pre-defined membership function to a crisp fuzzy output value.

Table 10. Fuzzy rules optimised by SDA and BFA.

No	SDA				BFA			
	Angle	Velocity	Torque	Weight	Angle	Velocity	Torque	Weight
1	PH	ZE	PH	0.4	PH	ZE	PH	0.6
2	NH	NH	ZE	0.6	NL	NL	ZE	0.8
3	NL	PL	PL	0.6	NL	PH	PL	0.6
4	PH	NL	ZE	0.2	PL	NL	ZE	0.4
5	PH	PH	PH	0.4	PL	PL	PH	0.4
6	ZE	PL	NL	0.8	ZE	PL	NL	0.8
7	PL	NL	PH	0.4	ZE	NL	PL	0.4
8	NL	PL	PL	0.6	NL	ZE	PL	0.6
9	NL	NL	PL	1	ZE	NL	PL	0.8
10	PH	NL	PL	0.6	PL	NL	ZE	0.6
11	PL	NH	NL	0.4	PL	NL	NL	0.4
12	PH	PL	NH	0.1	PL	PL	NH	0.8
13	NH	ZE	NL	1	NL	ZE	NL	0.8
14	NL	NL	ZE	0.4	NL	ZE	NL	0.4
15	NL	NH	PL	1	NL	NL	PL	0.8
16	NL	NL	NH	0.6	NL	NL	NL	0.6
17	PL	ZE	PL	0.8	PL	ZE	PL	0.8
18	NL	NL	ZE	0.8	NL	ZE	ZE	0.8
19	NL	NL	ZE	0.6	NL	ZE	ZE	0.6
20	NH	NH	PL	0.6	NL	NL	ZE	0.6
21	PL	NL	ZE	1	PL	ZE	PL	0.8
22	NH	NH	NL	0.6	NL	NL	NL	0.6
23	NH	NL	NL	0.4	NH	ZE	NL	0.6
24	PL	PL	NH	0.4	ZE	PL	NL	0.6
25	NL	PL	ZE	0.4	NL	PL	ZE	0.4

SDA: spiral dynamic algorithm; BFA: bacterial foraging algorithm.

HSBFA-based for PD-FLC optimisation

The effectiveness of the proposed HSBFA algorithm is validated by optimising the PD-like FLC for the flexible manipulator system. A single-input and multi-output state-space equation of motions presented in Refs. 30 and 35 is used to represent the flexible manipulator system. In particular, the HSBFA method is used to find the optimal control parameters for the PD-like FLC to result in better tracking and control performance. The proposed HSBFA algorithm is employed to optimise the proportional k_1 and derivative k_2 controller gains, fuzzy output gain k_3 , fuzzy sets, and the firing weight of the FLC system. Specifically, a total of 103 controller parameters are determined which results in a 103-dimensions problem. It consists of 75 fuzzy sets, 25 fuzzy firing weights, 1 proportional gain, k_1 , 1 derivative gain, k_2 , and 1 fuzzy output scaling factor, k_3 . The fuzzy set of the linguistic variables P , Q , and R that are denoted by $\{NH, NL, ZE, PL, PH\}$ as shown in equation (3) is represented by an integer value $\{1, 2, 3, 4, 5\}$, respectively. The fuzzy firing weights, proportional, derivative, and output gains are set as a value between $[0, 1]$. There are two search ranges defined in the algorithm. The first search range is between $[1, 5]$ and it is associated with the fuzzy sets. The second search range is between $[0, 1]$ and it is associated with fuzzy firing weights and gains of the hybrid controller. All the parameters are optimised simultaneously and they are configured in the optimisation algorithm such that the value is optimal if the error function of the algorithm is minimised.

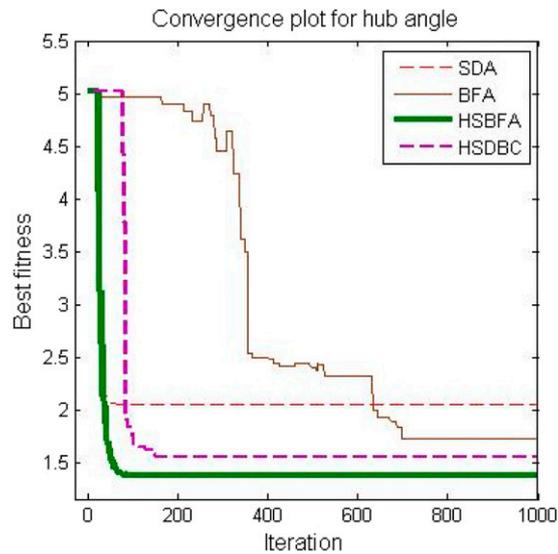
The error function for hub angle tracking employed in the optimisation algorithms is defined as the absolute mean error J , given by equation (4)

$$J = \frac{1}{N} \sum_{i=1}^N (|\theta_{desired(i)} - \theta_{actual(i)}|) \quad (4)$$

Table II. Fuzzy rules optimised by HSBFA and HSDBC.

No	HSBFA				HSDBC			
	Angle	Velocity	Torque	Weight	Angle	Velocity	Torque	Weight
1	NL	ZE	NL	1	PH	NH	ZE	0.3
2	NL	NH	NH	0.6	ZE	NL	PL	0.9
3	NH	PL	ZE	0.8	ZE	PL	NL	0.3
4	PL	PH	NL	0.4	PL	NH	NL	0.3
5	NH	PL	PH	0.6	PL	ZE	PL	0.2
6	ZE	NL	PL	0.6	PL	NL	ZE	0.2
7	NL	PH	ZE	0.4	NH	NL	NH	0.3
8	NH	NH	ZE	0.6	PL	PH	PL	0.4
9	NL	PH	PH	0.6	PL	PL	NL	0.1
10	PL	ZE	PL	0.6	ZE	PL	NL	0.9
11	NH	PH	PH	0.2	NL	NL	ZE	0.5
12	NL	PL	PH	0.8	NH	PL	PL	0.3
13	ZE	NH	PL	0.8	PL	NL	PH	0.9
14	PL	PL	PL	0.4	PL	NH	ZE	0.5
15	ZE	PL	NL	0.2	PH	NH	NL	0.9
16	NL	NL	PL	0.8	PH	PH	ZE	0.5
17	NL	NL	NL	0.6	NL	ZE	NL	0.7
18	PL	NH	NL	0.8	NL	PL	PH	0.4
19	PH	NL	NH	0.6	ZE	NH	NL	0.7
20	PL	ZE	PH	0.4	NL	ZE	ZE	0.1
21	NL	ZE	NH	0.4	ZE	NH	PH	0.2
22	PL	NH	ZE	0.4	NH	NL	ZE	0.9
23	NL	ZE	PL	0.2	NH	ZE	NL	0.6
24	ZE	PL	NL	0.6	PL	NH	ZE	0.9
25	PH	PL	PL	0.4	NL	ZE	NL	0.4

HSBFA: hybrid spiral-bacterial foraging algorithm; HSDBC: hybrid spiral dynamic bacteria chemotaxis.

**Figure 6.** Convergence plot of the algorithms.

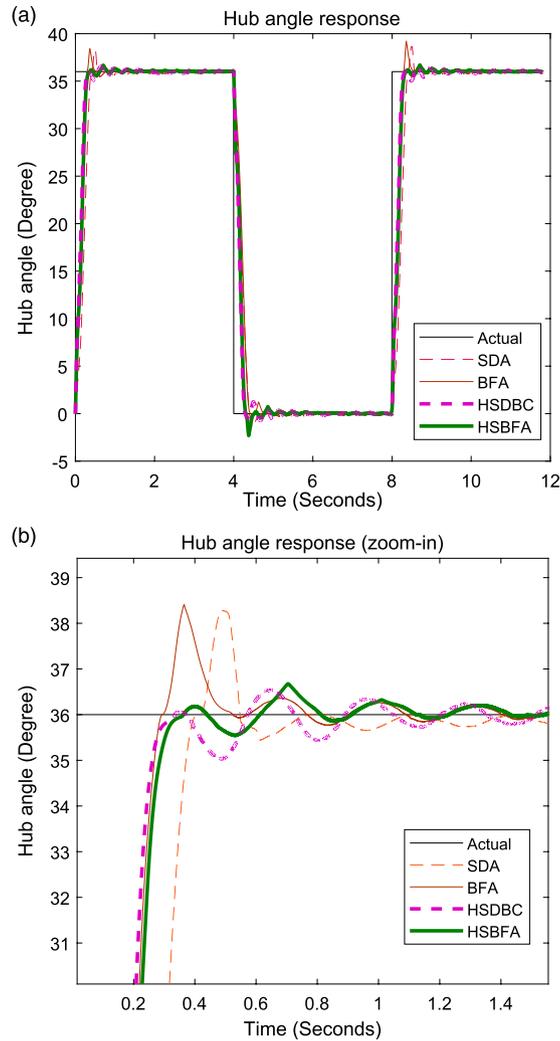


Figure 7. Time-domain response of the hub angle: (a) overall plot and (b) Zoomed-in for the first 1.4 s.

where the total number of data samples N is equal to 59,000 and $(\theta_{desired(i)} - \theta_{actual(i)})$ is the error between the desired and actual signal as shown in Figure 2. The advantage of this equation is that the accumulation of positive error values with a maximum of N data samples can be computed. As a result, a more accurate solution can be obtained. In the proposed HSBFA, the absolute mean error J is known as the agent's fitness.

The BFA, SDA, HSDBC, and HSBFA parameter values used in this controller optimisation study are depicted in Table 8. After performing exhaustive testing to achieve optimum performance, the total iteration $iter_{tot}$ for the algorithms was defined as 1000. The simulation was set to stop after the algorithms showed insignificant convergence behaviour in finding a better solution. All other critical parameters of the algorithms such as the radius, step size, and angle for the spiral model were determined based on a trial and error approach.

Table 9 shows the gain values of the optimised PD-like FLCs. Originally, the universe of discourse for the fuzzy inputs and output had a similar range within $[-1, 1]$ as shown in Figure 5. However, the original range of the membership function shown in Figure 5 was varied by multiplying it with the optimised gain values presented in Table 9. It is noted that the gains of the fuzzy inputs and fuzzy output optimised by the algorithms had different values. Hence, the new range of the universe of discourse affected the performance of the fuzzy logic controller.

The generated fuzzy rules for the optimised PD-like FLC are shown in Tables 10 and 11. Note that the same set of linguistic variables was obtained for rules 3 and 8 for the SDA algorithm. Similarly, the same set of linguistic variables are noted in rules 14, 18, and 19, but they produced different weights: 0.4, 0.8, and 0.6, respectively. For such a case, rule 18 with the highest weight is chosen, while rules 14 and 19 with lower weights are ignored. As a result, only 22 effective rules

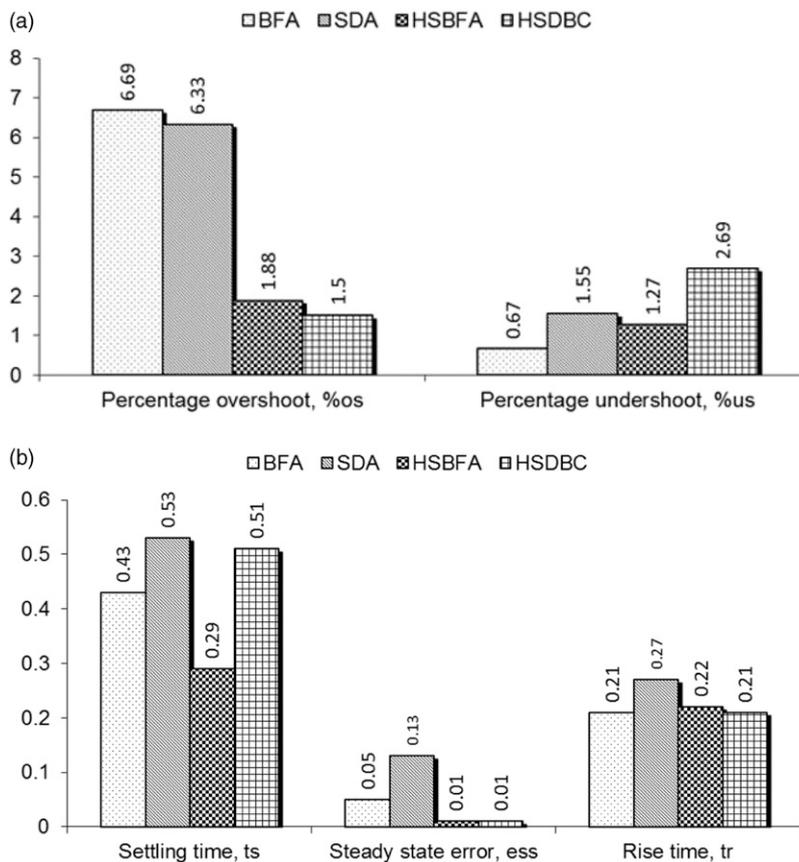


Figure 8. The performance of the flexible manipulator in a bar chart.

for the SDA-based FLC are produced. For the BFA-based FLC as shown in Table 10, rules 15 and 24 share the same set of linguistic variables with weights 0.2 and 0.6, respectively. Taking into account rule 24 and ignoring rule 15, the total effective number of rules for BFA-based FLC will be 24 rules. For HSBFA-based FLC (as shown in Table 11), rules 16 and 22 and rules 17 and 21 shared the same set of linguistic variables with the same weight. Furthermore, it shows that HSBFA-based FLC had higher redundancy of linguistic variables set with different weights than other algorithms. In particular, rules 2 and 20, rules 4 and 10, rules 7 and 9, rules 6 and 24, rules 18 and 19, and rules 13 and 14 produced the same set of linguistic variables but with different weights. Therefore, the total number of effective rules for the HSBFA-based FLC will be 17 by ignoring rules 4, 7, 14, 19, 20, 21, 22, and 24. For HSDBC-based FLC as shown in Table 11, rules 3 and 10, rules 17 and 25, and rules 14 and 24 shared the same set of linguistic variables, but they produced different weights. Considering rules 10, 17, and 24 and ignoring rules 3, 14, and 25, the total number of effective rules for the HSDBC-based FLC will be 22.

The convergence responses for tuning the PD-like FLC parameters are shown in Figure 6. This consists of the convergence plot for the four algorithms in terms of iteration. Notice that at the early iterations, the SDA showed the fastest speed followed by HSBFA, HSDBC, and BFA. However, after 30 iterations the trend changed and HSBFA successfully intercepted the SDA graph. The SDA graph started to gradually converge to a local optimum while the HSBFA sharply converged until iteration 40 before converging gradually to a nearly global optimum point. The HSBFA achieved the best location among the four algorithms under study. The HSDBC hardly converged at the beginning of the search operation until it reached iteration 76. However, between iterations 77 and 105, it converged sharply. It showed a slower convergence starting from iteration 106, and it finally got trapped in a local optimum location. The BFA showed the worst performance in terms of convergence speed where it started to converge at the beginning of iteration 162 and was the slowest among the four algorithms. However, the BFA successfully intercepted the SDA after 633 iterations, and it was unable to converge further after 702 iterations.

The responses of the hub angle position of the flexible manipulator optimised by the algorithms are shown in Figure 7(a). Figure 7(a) portrays an overall picture of the response while Figure 7(b) shows a zoomed-in graph for the first 1.4 s of the

response. Figure 8 shows the control performance analysis of the hub angle response computed based on the results in Figure 7(b). The results clearly show that the BFA-based system response had the highest percentage overshoot while the HSDBC-based system response had the highest percentage undershoot with 6.69% and 2.69%, respectively. It is observed that the BFA-based and HSDBC-based system responses shared the lowest rise time followed by HSBFA and SDA with the achieved times of 0.21 s, 0.22 s, and 0.27 s, respectively. Here, the hybrid controller optimised by HSBFA showed a very good performance where it could produce significantly low overshoot and undershoot although the rise time was almost similar to the fastest. For the steady-state error, the BFA and SDA achieved the second and third places with the errors 0.05 and 0.13, respectively while the HSBFA and HSDBC performed significantly better than the two algorithms with the same error of 0.01. In terms of the settling time, as depicted in Figure 7(a), the HSBFA showed the best achievement where the response was kept within 2% of the final point (i.e. 36 degrees). The BFA was in second place after HSBFA at the time of 0.43 s followed by HSDBC and SDA at 0.52 and 0.53 s, respectively. The performances are shown in Figures 7 and 8 comply with the results shown in Figure 6 where overall, the HSBFA showed the best performance followed by HSDBC, BFA, and SDA. The HSBFA had a balanced trade-off considering all the five performance specifications. The HSBFA has the best performance for the settling time and steady-state error while there is no worst performance for the HSBFA. The HSDBC has the best performance for the steady-state error, rise time, and overshoot, but it has a significantly worst performance for the undershoot. Noted that there is only a small difference between the rise time and overshoot performances of HSBFA and HSDBC. This gives an advantage to the HSBFA and hence outperformed the HSDBC and the other two algorithms.

It is also noted from Figure 8 that the HSBFA-based and HSDBC-based system has shown considerable performance difference in terms of percentage overshoot. The chart also shows a significant difference in the control performance of the settling time between HSBFA-based and the other systems. Among the five characteristics of the time-domain response, the SDA-based system had the worst control performance in terms of steady-state error, settling time, and rise time. The BFA-based and HSDBC-based systems had the worst control performance in terms of percentage overshoot and percentage undershoot, respectively. The chart depicts that the difference is significantly large compared to the others. None of the HSBFA-based system time-domain response characteristics had the worst control performance than the other three systems. Overall, the best control performance for the hub angle position control of the flexible manipulator system was produced by the HSBFA-based system where it achieved a balanced combination of the dynamic response parameters.

Conclusion

A slow convergence rate is the main deficiency of BFA. On the contrary, SDA has a faster convergence rate, but the likelihood that it will be trapped in a local optimal solution is relatively high especially in the case of higher-dimensional problems. In this paper, to tackle the problems, a synergy between the BFA and SDA has been proposed. A spiral model of the SDA has been incorporated into the chemotaxis phase of the BFA where the movement of the bacteria is guided by the best bacterium found in the previous iteration and the synergy combines the random and spiral deterministic approaches in the motion. Moreover, the combination of the spiral model into the chemotaxis phase of BFA has introduced bacterium with a dynamic step size that enhances the algorithm to produce the near-global optimal solution. Hence, the possibility of locating the near-global optimal solution at any location in the search space has increased.

The proposed algorithm has been validated with a set of state-of-the-art benchmark functions comprising 10 and 30 dimensions taken from the CEC2013. The results have been compared with those of predecessor algorithms, that is, BFA, SDA, and other types of optimization algorithms. The result of the test has been statistically analysed through the non-parametric Friedman and Wilcoxon signed-rank analytical approaches to determine the best algorithm that can locate a position close to the theoretical global optimum point. Moreover, in order to show the effectiveness of the proposed algorithms in solving a complex and real-engineering problem, they have been employed to automatically obtain and optimise a nonlinear composite intelligent FLC for tracking control of hub angle position of a flexible manipulator system.

The performance tests and analyses carried out with benchmark functions have shown that the HSBFA has produced better accuracy compared to the other contested algorithms and it has been ranked as the first out of the five algorithms. On the other hand, the results of tests on the flexible manipulator system have shown that the proposed hybrid algorithm has optimally acquired the FLC control parameters and has tracked the hub angle better compared to its predecessor and other well-known algorithms.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The work has been financially sponsored by the Fundamental Research Grant Scheme FRGS/1/2019/ICT05/UMP/03/1 and awarded by the Ministry of Higher Education Malaysia (MOHE) via Research and Innovation Department, UMP. It has the university reference number RDU1901217.

ORCID iD

Ahmad Nor Kasruddin Nasir  <https://orcid.org/0000-0002-4133-5558>

References

1. Shawky A, Zydek D, Elhalwagy YZ, et al. Modelling and nonlinear control of a flexible-link manipulator. *Appl Math Model*2013; 37(23): 9591–9602.
2. Zhao X, Shi P, and Zheng X. Fuzzy adaptive control design and discretization for a class of nonlinear uncertain systems. *IEEE TransCybernetics*2016; 46(6): 1476–1483.
3. Lai G, Liu Z, Philip CL, et al. Adaptive fuzzy tracking control of nonlinear systems with asymmetric actuator backlash based on a new smooth inverse. *IEEE TransCybernetics*2016; 46(6): 1250–1262.
4. Zhao X, Shi P, Zheng X, et al. Intelligent tracking control for a class of uncertain high-order nonlinear systems. *IEEE TransNeural NetwLearn Syst*2016; 27(9): 1976–1982.
5. Lai G, Liu Z, Philip Chen CL, et al. Asymmetric actuator backlash compensation in quantized adaptive control of uncertain network nonlinear systems. *IEEE TransNeural NetwLearn Syst*2017; 28(2): 294–307.
6. Chen C, Liu Z, Zhang Y, et al. Modelling and adaptive compensation of unknown multiple frequency vibrations for the stabilization and control of an active isolation system. *IEEE TransControl Syst Technology*2016; 24(3): 900–911.
7. Masrom MF, Ghani NMA, and Tokhi M. Particle swarm optimization and spiral dynamic algorithm-based interval type-2 fuzzy logic control of triple-link inverted pendulum system: a comparative assessment. *JLow Frequency Noise, Vibration Active Control*2021; 40(1): 367–382.
8. Hadi MS, Darus IZM, Tokhi MO, et al. Active vibration control of a horizontal flexible plate structure using intelligent proportional–integral–derivative controller tuned by fuzzy logic and artificial bee colony algorithm. *JLow Frequency Noise, Vibration Active Control*2020; 39(4): 1159–1171.
9. Anh HPH and Ahn KK. Hybrid control of a pneumatic artificial muscle (PAM) robot arm using an inverse NARX model fuzzy model. *Eng ApplArtif Intelligence*2011; 24: 697–716.
10. Mandal P, Sarkar BK, Saha R, et al. Real-time fuzzy feedforward controller design by bacterial foraging optimization for an electrohydraulic system. *Eng ApplArtif Intelligence*2015; 45: 168–179.
11. Banerjee S, Chakrabarty A, Maity A, et al. Feedback linearizing indirect adaptive fuzzy control with foraging based on-line plant model estimation. *Appl Soft Comput*2011; 11(4): 34413450.
12. Roy R, Bhatt P, and Ghoshal SP. Evolutionary computation based three-area automatic generation control. *Expert SystAppl*2010; 37(8): 5913–5924.
13. Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst Mag*2002; 22(3): 52–67.
14. Dasgupta S, Das S, et al. Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. *IEEE TransEvol Comput*2009; 13(4): 919–941.
15. Mishra S. Hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE TransEvol Comput*2005; 9(1): 61–73.
16. Kim DH, Abraham A, and Cho JH. A hybrid genetic algorithm and bacterial foraging approach for global optimization. *JInfSci*2007; 177(18): 3918–3937.
17. Kim DH and Cho JC. Advanced bacterial foraging and its application using fuzzy logic based variable step size and clonal selection of immune algorithm. Proceedings of 2006 International Conference on Hybrid Information Technology (ICHIT'06), Ceju Island, 9–11 November 2006, pp. 293–298.
18. Biswas A, Dasgupta S, Das S, et al. Synergy of PSO and bacterial foraging optimization - a comparative study on numerical benchmarks. *InnovHybrid Intell Syst*2007; 44: 255–263.
19. Panigrahi K and Pandi VR. Bacterial foraging optimisation: Nelder-Mead hybrid algorithm for economic load dispatch. *IET Generation, Transm Distribution*2008; 2(4): 556–565.
20. Tamura K and Yasuda K. Spiral multipoint search for global optimization. Proceeding of 10th International Conference on Machine Learning and Applications, Honolulu, USA, 18–21 December 2011, pp. 470–475.

21. Nasir ANK, Tokhi MO, Ghani NMA, et al. A novel hybrid spiral-dynamics bacterial-foraging algorithm for global optimization with application to control design. Proceeding of the 12th UK Workshop on Computational Intelligence (UKCI2012), Edinburgh, UK, 5–7 September 2012, pp. 1–7.
22. Nasir ANK and Tokhi MO. A novel hybrid bacteria chemotaxis spiral-dynamic algorithm with application to modelling of flexible systems. *Eng Appl Artif Intelligence* 2014; 33(1): 31–46.
23. Nasir ANK, Tokhi MO, Ghani NMA, et al. A novel hybrid spiral dynamics bacterial chemotaxis algorithm for global optimization with application to controller design. Proceedings of the International Conference on Control (UKACC 2012), Cardiff, UK, 3–5 September 2012, pp. 753–758.
24. Nasir ANK and Tokhi MO. A novel hybrid spiral-dynamics random-chemotaxis optimization algorithm with application to modelling of a flexible robot manipulator. Proceeding of 16th International Conference on Climbing and Walking Robots (CLAWAR2013), 14–17 July 2013, Sydney, Australia, pp. 667–674.
25. Nasir ANK and Tokhi MO. An improved spiral dynamic optimization algorithm with engineering application. *IEEE Trans Syst Man, Cybernetics: Syst* 2015; 45(6): 943–954.
26. Nasir ANK, Tokhi MO, and Abd Ghani NM. Novel hybrid bacterial foraging and spiral dynamics algorithms. 13th UK Workshop on Computational Intelligence (UKCI2013), Surrey, UK, 9–11 September 2013, pp. 199–205.
27. Hooke R and Jeeves TA. Direct search solution of numerical and statistical problems. *J Assoc Comput Machinery* 1961; 8(2): 212–229.
28. Liang JJ, Qu B-Y, Suganthan PN, et al. Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. In: *Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*; January 2013.
29. Remya K, Varun P, and Prakash K. Performance of particle swarm optimization and pattern search on CEC 2013 real parameter single objective optimization. 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI2015), Kanpur, India, 14–17 December 15, pp. 1–6.
30. Alam MS and Tokhi MO. Hybrid fuzzy logic control with genetic optimisation for a single-link flexible manipulator. *Eng Appl Artif Intelligence* 2008; 21(6): 858–873.
31. Azad AKM. Analysis and design of control mechanisms for flexible manipulator systems. PhD Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK; 1994.
32. Electrocraft Corporation. *DC Motors Speed Control Servo Systems*. Minnesota: Electrocraft Corporation / Robbins & Mayers, 1985.
33. Elliott JR, Dubay R, Mohany A, et al. Model predictive control of vibration in a two flexible link manipulator — part 2. *J Low Frequency Noise, Vibration Active Control* 2014; 33(4): 469–483.
34. Bian Y and Gao Z. Nonlinear vibration control for flexible manipulator using 1: 1 internal resonance absorber. *J Low Frequency Noise, Vibration Active Control* 2018; 37(4): 1053–1066.
35. Tokhi MO and Mohamed Z. Finite element approach to dynamic modelling of a flexible robot manipulator: performance evaluation and computational requirements. *Commun Numer Methods Eng* 1999; 15: 669–676.