Fusion Bacterial Foraging and Particle Swarm Optimization Algorithm For An Autonomous Hovercraft Control

Journal:	*Part C: Journal of Mechanical Engineering Science*
Manuscript ID	JMES-22-2936.R3
Manuscript Type:	Original Research Article
Date Submitted by the Author:	31-Jul-2023

Complete List of Authors:	Tran, Khoa; Tamkang University College of Business and Management Pham Duc, Lam; Faculty of Engineering and Technology, Nguyen Tat Thanh University HCM, VN Tran, Thanh Trang; Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Food Industry Nguyen, Hoang Nam; Modeling Evolutionary Algorithms Simulation and Artificial Intelligence, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University

Keywords:	Control Engineering, Automobile, Motion Control, Vehicles, Particle Swarm Optimization

Abstract:	This article proposes a novel method for designing an effective autonomous hovercraft vehicle controller. In the first step, we design the proportional (P) and derivative (D) gains of a PD like fuzzy controller, in which P and D parameters are used as the fuzzy scaling factors of the controller design. Next, bacterial foraging (BF) and particle swarm optimization (PSO) are combined to create a fusion algorithm to optimize the P and D gains of the PD like fuzzy controller. The integral of time multiplied square error (ITSE) fitness function is applied to find the minima of the controller design error criterion. Numerical simulations are then implemented on a real autonomous hovercraft motion model. Results indicate that the proposed controller design is more stable, reliable and accurate than existing controllers.

Fusion Bacterial Foraging and Particle Swarm Optimization Algorithm For An Autonomous Hovercraft Control

Huu Khoa Tran[1], Pham Duc Lam[2], Tran Thanh Trang[3], and Hoang Nam Nguyen[4,*]

[1] Department of Information Management, Tamkang University, New Taipei City, Taiwan; email: hk5tran@mail.tku.edu.tw, https://orcid.org/0000-0001-5268-2355

[2] Faculty of Engineering and Technology, Nguyen Tat Thanh University, Vietnam. email: pdlam@ntt.edu.vn

[3] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Food Industry, Ho Chi Minh City, Vietnam; trangtt@hufi.edu.vn

[4] Modeling Evolutionary Algorithms Simulation and Artificial Intelligence, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, 19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam;

* Correspondence: nguyenhoangnam@tdtu.edu.vn

Abstract: This article proposes a novel method for designing an effective autonomous hovercraft vehicle controller. In the first step, we design the proportional (P) and derivative (D) gains of a PD like fuzzy controller, in which P and D parameters are used as the fuzzy scaling factors of the controller design. Next, bacterial foraging (BF) and particle swarm optimization (PSO) are combined to create a fusion algorithm to optimize the P and D gains of the PD like fuzzy controller. The integral of time multiplied square error (ITSE) fitness function is applied to find the minima of the controller design error criterion. Numerical simulations are then implemented on a real autonomous hovercraft motion model. Results indicate that the proposed controller design is more stable, reliable and accurate than existing controllers.

Keywords: Bacterial Foraging (BF) Algorithm, Particle Swarm Optimization (PSO), PD like Fuzzy, Hovercraft, Integral of Time Multiplied Square Error (ITSE).

1. Introduction

In recent years, many researchers have focused on using nonlinear control to build an autonomous hovercraft [1–6] that can work well on various surfaces such as water, sand beach, ice and especially on muddy land. Most of these available controllers of the real hovercraft models are just based on the traditional PD controller and very few methods of optimizing the controller parameters are introduced for this system. The main aim of this paper is to propose a new way to optimize the proportional and derivative gains of the PD like fuzzy controller of the hovercraft.

Based on an expert knowledge system, fuzzy logic controllers (FLC) have the advantage of solving problems by using flexible if-then rules. Many works [7-14] have utilized this technique to solve various control system topics and achieved significantly better results than other methods. Hence, this research also applies this approach to the control system of hovercraft models. In this paper, we design a set of fuzzy triangular membership functions and optimizes scaling factors (SF) as the inputs of fuzzy controller. Although evolutionary algorithms [15, 16] such as particle swarm optimization (PSO), simulated annealing (SA) and genetic algorithm (GA), can easily detect the local minima but it is difficult to detect the global optimization. PSO algorithms [17-21] are a population-based heuristic algorithm that is invented based on observing animal swarm social behavior to detect accurate targets in multi-dimensional space. PSO uses individual particles to perform searches. To finding the optimal solution, each particle decides its search direction based on its best previous location (cognitive part) and all other members' best locations (social part). In 2002, K. M. Passino [22], inspired from the behavior of *Escherichia-coli* bacteria, firstly proposed a bionic algorithm that is very effective for distributed optimization and control design. Although, BF algorithm is very effective in local search, it is easily tapped or delayed in reaching global solution because of the random search directions during chemotaxis process that is the movement of an organism or entity in response to a chemical stimulus.

Although, BF algorithm is very effective in local searches, it is easily tapped or delayed in reaching global solution because of the random search directions during the movement of an organism or entity in response to a chemical stimulus called chemotaxis process.

To avoid local optima traps and improve the optimal performance in engineering application research, several PSO-based algorithms are used to solve the problems, for instance: GA and PSO [23], PSO and SA [24], hybrid PSO-Differential Evolution [25], Direct Search method and hybrid PSO [26], hybrid swarm intelligence approach [27]. Also, combining PSO with BF has emerged as one of the most efficient optimization methods, which can be applied successfully in many fields, including mathematical optimization [28], RFID network planning [29], medical care [30] and traditional PID control [31-36] in electric power systems. The above techniques can expand the optimal convergence pace by taking the advantages while eliminating the weakness of the solo algorithms.

Thus, we employed a fusion algorithm, called fusion bacterial foraging and particle swarm optimization (named hereafter fBF-PSO), to increase the convergence speed in shorter operating generation and adjust the global search efficiency. We use PD like fuzzy control to improve hovercraft system stability. The fBF-PSO technique is also used to improve the performance of this PD like fuzzy control by minimizing the ITSE fitness cost function [37-39]. The proposed controller can obtain high-performance control of the Hovercraft in terms of higher accuracy, quick response, high stability and smoother maneuvers. To the best of our knowledge, this is the first paper that combines the advantages of PSO and BF method in designing an effective and optimal PD like fuzzy controller for a real hovercraft model. In section 2, we describe the hovercraft configuration. In section 3, optimization algorithm and control strategy are introduced. In section 4, numerical simulation control of a real hovercraft model is presented. Finally, conclusions are discussed in section 5.

2. Autonomous Hovercraft configuration

The air cushion vehicle, which is usually called as hovercraft, is composed of rotors and a cushion. Blower (rotor duct fan) is used to provide a large volume of air inside the air cushion, which is larger than the atmospheric pressure. Therefore, the hovercraft can float and is capable of travelling over ice, water, land, mud… [1-6, 13]. A tilt servo motor, which is located at the rear, is usually employed to steer the hovercraft. Although many modern technologies are used [1-6, 13], there is still a need for a more advanced hovercraft maneuvering system. Especially, faster response and higher reliability are of great interest. Figure 1 shows a popular hovercraft model used in this research, which has a single tilt servo motor, a propeller settled along z-axis and a blower attached along y-axis.

The dynamic hovercraft model in this paper is referenced from [1, 6 and 13] with right hand coordinate systems. X-axis is used to position in the lateral direction whilst Y-axis is along its body. As shown in the figure 1, the Z-axis is vertical and its positive direction is downward. The hovercraft's kinematics can be briefly described as follow:

$$\begin{cases} \dot{x} = p\cos\varphi - s\sin\varphi \\ \dot{y} = q\cos\varphi - s\sin\varphi \\ \dot{\varphi} = \omega \end{cases} \tag{1}$$

where $\boldsymbol{\omega} \in \boldsymbol{R}$ represents the angular velocity. $\boldsymbol{p},\boldsymbol{q},\boldsymbol{s} \in \boldsymbol{R}$ are defined as linear velocities in surge direction, sway direction and steering, respectively. From this above equation, we can derive the kinetic and potential energies of the hovercraft to compute L'agrange $L=T\text{-}V$. Then we can apply Euler-Lagrange formulation on equation 2:

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{q} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{T}_q \\ \boldsymbol{0} \end{bmatrix} \tag{2}$$

where $\boldsymbol{T_q} \in \boldsymbol{R}$ represents the torque in yaw and $\boldsymbol{F} \in \boldsymbol{R}$ is the control force in the surge direction. The torque control, which is perpendicular to the center of the hovercraft propeller, is a function of $\boldsymbol{F} \in \boldsymbol{R}$.

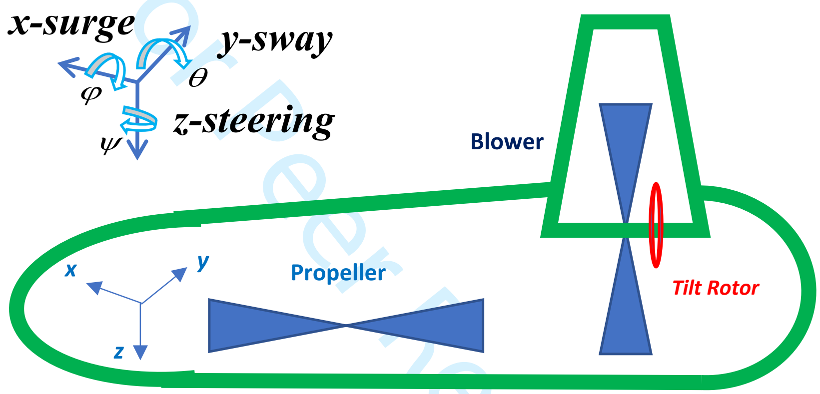


Figure 1. The Hovercraft prototype model.

3. Optimization Algorithms and Control Strategy

3.1.1. PD controller

The PD controller, which is implemented in this article, is high efficiency despite having a simplistic structure. Its design can reduce the settling time whereas improve the system stability. For the ith area, the PD controller combined gain, $K(s)$, is computed from the following equation 3:

$$\boldsymbol{G}_{(s)} = \boldsymbol{K_P} + \boldsymbol{K_D s} \tag{3}$$

Where, K_P and K_D are the conventional PD controller proportional and derivative gains, respectively.

3.1.2. Fuzzy Logic Controller (FLC)

The FLC dynamic behavior are based on a set of linguistic rules and originated from expert knowledge [7-9]. Designer needs to decide the input and output variables for building a suitable set of fuzzy rules. In this article, the error *e(t)* and the error rate *de(t)* are chosen as input variables, while the control output is also signal input to the system and considered as c_i. Then, the relationship between these two inputs and one output variables are investigated. The error and error rate as well as the fuzzy rules determine the online change of the system output c_i. We then need to fuzzify and defuzzify *e(t), de(t)* and c_i parameters. Fig. 2 illustrate the fuzzy inference system employed in this work. The center of area method defuzzification and Mamdani's MIN–MAX inference engine is utilized. Seven linguistic triangular membership functions for the two inputs and one output are assigned: positive big PB, positive medium PM, positive small PS, zero ZE, negative small NS, negative medium NM, and negative big NB. Table 1 explains the Fuzzy controller rules.

3.1.3 PD like Fuzzy controller

An intelligent control apply to autonomous hovercraft is the PD like Fuzzy controller. In that, FLC could be used as a classical PD controller, and overcome the disadvantages of the PD Controller. It is necessary to select the input and output variables and proper controller rules, as illustrated on equation 4:

$$\boldsymbol{u_{(t)} = K_P e_{(t)} + K_D de_{(t)}} \tag{4}$$

Where *e(t)* is the error, and *de(t)* is the change in error. In the following subsections, the proposed optimization algorithms are presented to optimize the SF as inputs to a fuzzy controller.

Table 1. Rule base of Fuzzy Logic Controllers

Control signal CI(t)		***Input e(t)***						
		NB	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
Input de(t)	**NB**	**ZE**	PS	PM	PM	**PB**	**PB**	**PB**
	NM	NS	**ZE**	PS	PM	PM	**PB**	**PB**
	NS	NM	NS	**ZE**	PS	PM	PM	**PB**
	ZE	NM	NM	NS	**ZE**	PS	PM	PM
	PS	**NB**	NM	NM	NS	**ZE**	PS	PM
	PM	**NB**	**NB**	NM	NM	NS	**ZE**	PS
	PB	**NB**	**NB**	**NB**	NM	NM	NS	**ZE**

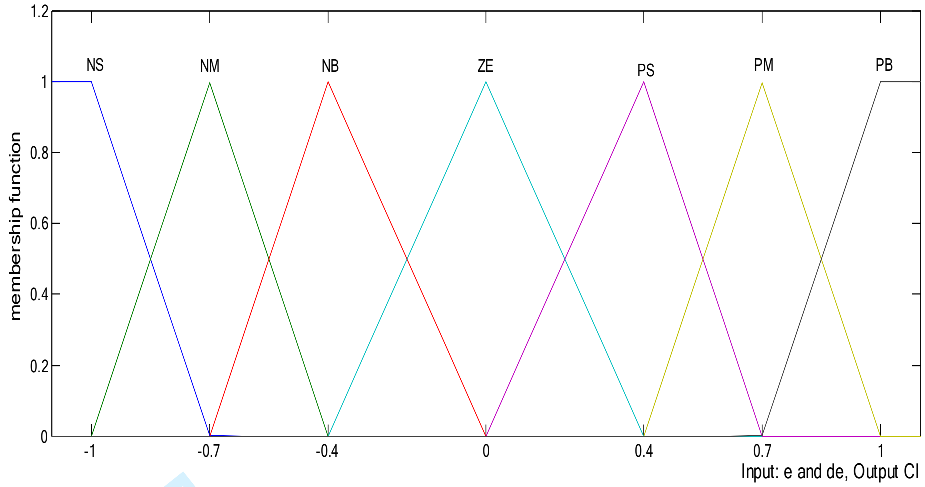


Figure 2. Fuzzy membership function.

3.2.1. PSO Algorithm

Recently, the PSO has emerged as the one of the most well-known and powerful tools for optimization [17-21]. We try to adapt PSO algorithm for the hovercraft control. Specifically, PSO algorithm is combined with another effective optimization algorithm to optimally tune the PD controller gains. In order to follow along with learning rates c_1 and c_2, the inertia weight □ is a user-defined parameter. It manages the relationship of the previous values of particle velocities to the current value. The PSO implementation process is illustrated by the flow chart in Fig. 3. The PSO mathematical equations (5) and equation (6) are briefly described as follows:

$$\boldsymbol{V(t+1) = \omega V(t) + c_1 r_1 (Pbest(t) - P(t)) + c_2 r_2 (Gbest(t) - P(t))} \quad (5)$$

$$\boldsymbol{P(t+1) = P(t) + V(t+1)} \quad (6)$$

Where V is particle velocity, P is the current position, Pbest is local best position and Gbest is global best position; ω is the inertia weighting factor, c_1 and c_2 as learning rates. The variables r_1 and r_2 are uniformly distributed random numbers [0, 1].

3.2.2. BF optimization

The BF optimization derives from searching for information on the Escherichia (E-coli) bacteria [22] capabilities to survive in the natural changing environment. During the evolution process, fitness criteria rely on their mobile behavior to maintain a proper foraging strategy as well as reshape or eliminate their poor strategy when they are searching for food. Bacteria with good foraging strategies are then replicated in the next generation by the evaluation chain. The E-coli bacteria foraging progression to global searching capability is simplify described by four significant steps of Chemo-tactic, Swarming, Reproduction and Elimination–Dispersal. First, Chemo-tactic (θ_k) process illustrates the E-coli bacteria motion by two different methods; one can swim until a fixed time is reached while the other can tumble to alternate between two lifetime modes of operation. Where, θ_k symbolizes the *k-th* bacterium, N_k is the size of the step taken in the random direction specified by the tumble and Δ is a length of the random direction unit vector, as shown in equation (7). Second, Swarming step (J_{cc}): bacteria move to follow the nutrient gradient produced by the nutrient consuming group. Bacteria will release an attractant when they reach high nutrient areas and form concentric patterns of higher density bacteria. Mathematically, swarming can be described by equation (8). Where q is the number of variables to be optimized, which are present in each bacterium and $\theta = [\theta_1, \theta_2, \ldots, \theta_q]^T$ is a point in the p-dimensional search domain; d_a and e_a are the attractive coefficients; h_r and e_r are repellant coefficients. Third, Reproduction (N_r) step**: t**o maintain the constant population, the healthier bacteria split into two and the less healthy bacteria die. As shown in equation (9), Sr is the number of population members that have had enough nutrients to reproduce (split in two) without mutation. Finally, Elimination and Dispersal step: may occur when local significant increases in heat kill bacteria that are currently in nutrient-rich regions. Bacteria can be dispersed by an abrupt flow of water. Elimination and dispersal events may destroy chemotactic progress, but they may also promote chemotaxis, since they place bacteria near food sources. Soon after BF algorithm was invented, it has been widely used for optimization in numerous science and technology fields.

$$\boldsymbol{\theta_k = \theta_k + N_k * \Delta_k} \tag{7}$$

$$\boldsymbol{J_c = -\, d_a exp\Big[-e_a \textstyle\sum_{m=1}^{q}\big(\theta_m - \theta_m^i\big)^2\Big] + h_r exp\Big[-e_r \textstyle\sum_{m=1}^{q}\big(\theta_m - \theta_m^i\big)^2\Big]} \tag{8}$$

$$\boldsymbol{S_r = S/2} \tag{9}$$

3.2.3. Fusion Bacterial Foraging-Particle Swarm Optimization algorithm

Several studies [28-36] combined the ability of PSO to exchange social information with the ability of BF to eliminate and disperse old problems into new solutions**.** This combination was called fBF-PSO, which has the combined strength of the two algorithms. To achieve optimal convergence, the proposed fBF-PSO algorithm is derived from the specified search directions of bacteria tumbling, which are oriented simultaneously by the individual and global best locations, and the identified ideas of PSO is updating position and velocity. This means that the proposed algorithm utilizes a PSO velocity calculation to precisely determine the updated chemotaxis drop direction within the BF algorithm. This allows tumble's unity-length random order to be adapted to the best global as well as the best individual position.

Several parameters must be evaluated, including: "iterations, inertia weight, learning rates, the number of bacteria, chemotactic steps, limit swimming distance, reproduction steps, elimination-dispersal events, and probability that each bacterium is eliminated". These are referred to in [16-19, 22, 28-36] and in our own experimental controller designs. Hence, we combine fBF-PSO and the PD like Fuzzy control for building the hovercraft controller. This algorithm is then implemented to find the SF as the best optimization parameters of the proposed controller. In the control diagram illustrated in Fig. 3, the tracking error *e(t)* and differential tracking error *de(t)* are employed as inputs of the fuzzy inference system. Two main parameters K_P (proportional gain) and K_D (derivative gain) of the PD like Fuzzy control are then optimally tuned with the integral of time multiplied square error (ITSE) performance index [37-39]. After this, the results are compared qualitatively and quantitatively in terms of the ITSE minimum value: the settling time, the rising time, the overshoot and the tracking error, as demonstrated in equation (9):

$$\boldsymbol{ITSE = \int_0^{\infty} e^2(t)\, t\, \mathrm{d}t} \tag{9}$$

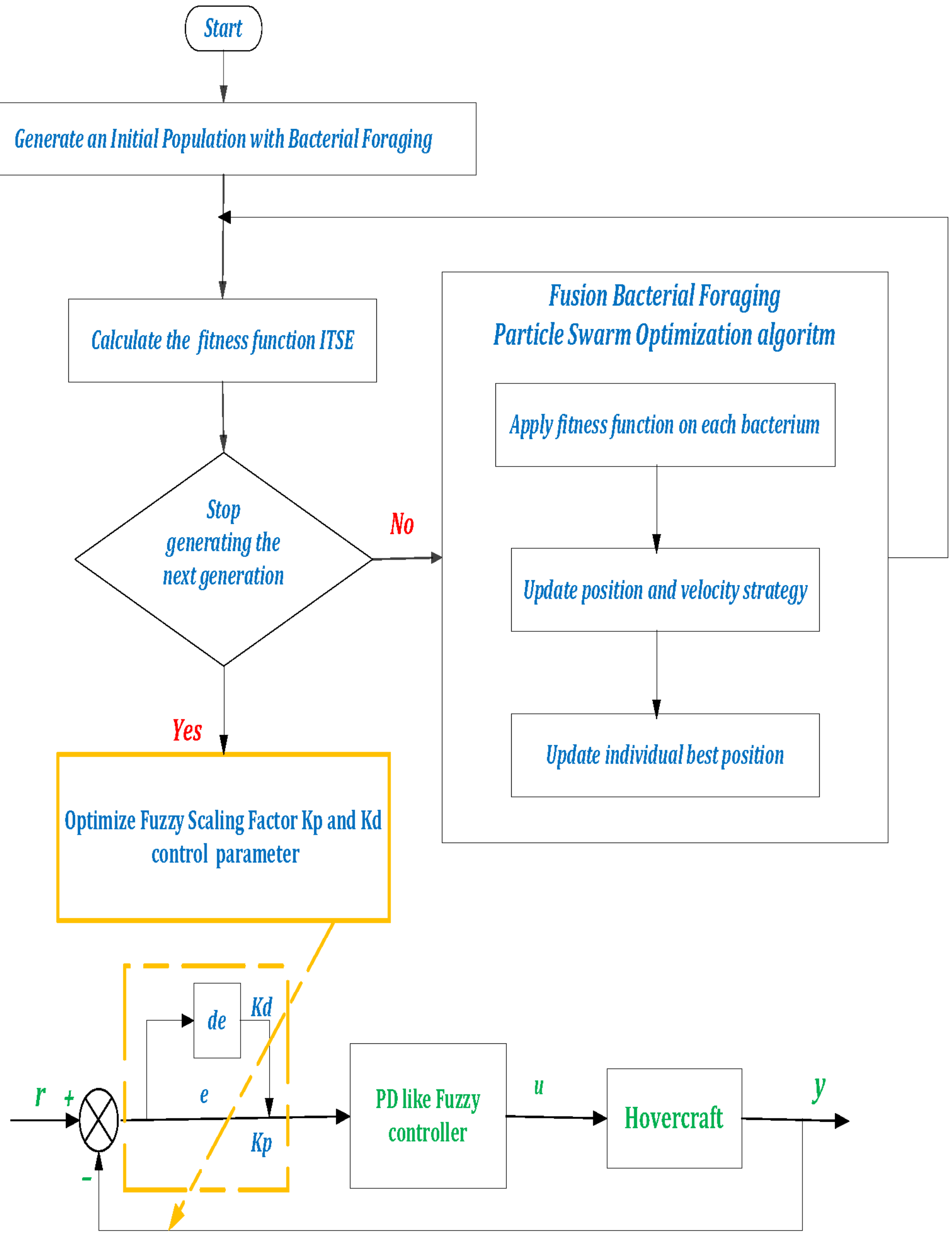


Figure 3. The fBF-PSO algorithm diagram.

4. Numerical Peformance

In this section, MATLAB/Simulink software is used to evaluate the performance of the proposed controller designs for autonomous Hovercraft mathematical models. These models were derived from the previous research [1, 6 and 13] with the mass of Hovercraft $m = 2.1\ kg$ and the inertia moment $I = 0.000257$. The simulation sampling time is 0.01 second. Numerical simulation results are operated in the stochastic optimization space. The proposed optimizations are simply set on 50 iterations. The PD gains of the proposed controllers are set in range ϵ [0, 100]. The significant optimal parameters of each optimization algorithm are chosen and displayed in the Table 2.

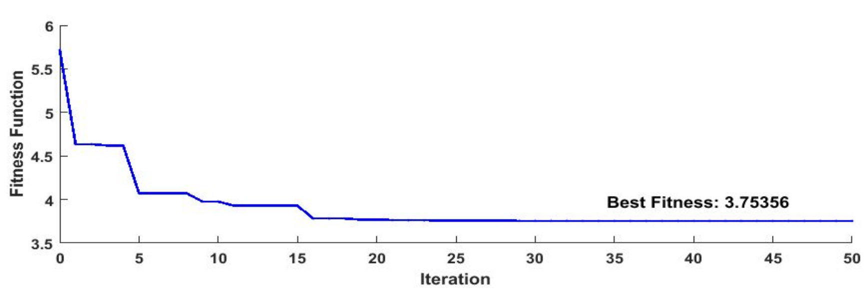
Table 2. Algorithms parameters setup

Parameters	PSO	BF	fBF-PSO
Dimension of search space	2	2	2
Generation or Iteration	50	50	50
Acceleration coefficient, c_1	1.22	-	1.22
Acceleration coefficient, c_2	1.22	-	1.22
Inertia weighting factor	0.92	-	0.92
Number of bacteria	-	50	50
Number of chemotactic steps	-	10	10
Limits the length of a swim	-	5	5
Number of reproduction steps	-	5	5
Number of elimination-dispersal events	-	3	3
Probability of each bacteria getting eliminated or dispersed	-	0.2	0.2

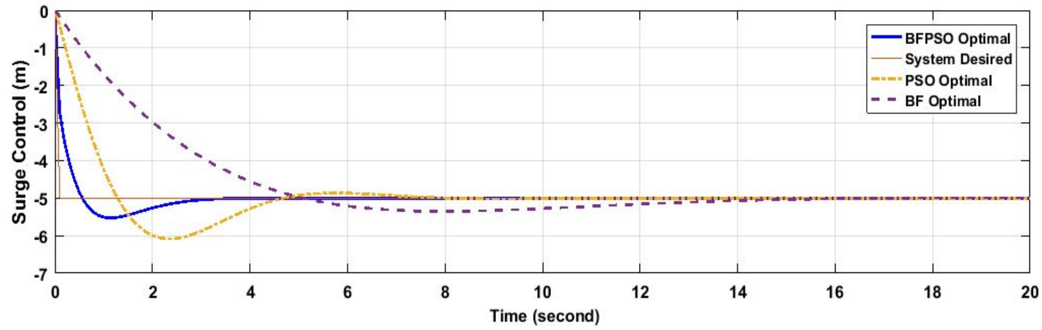
To compare the fBF-PSO with the standard BF and conventional PSO, this paper investigate the performances of three channels of autonomous hovercraft motions: surge position x in Figure 4, sway position y in Figure 5 and yaw steering angle in Figure 6, respectively. The numerical performances display that the fBF-PSO methodology achieved the significant stability and dominance in performance response compare to the solo BF or the single PSO algorithm. All presented results proved that fBF-PSO achieved the optimal control including: response time, less error, just after the third second at all, and were illustrated on the Table 3. Generally, all optimal controller designs achieve the high-quality performance in terms of stability, precision and reliability.

Table 3. The fBF-PSO proposed method tuning gain results

	Surge position x (5 m)	*Sway position y (5 m)*	*Steering-Yaw angle (10 degree)*
Generation	*50*	*50*	*50*
K_P	*78.324*	*45.783*	*34.569*
K_D	*50.671*	*88.123*	*77.257*
Settling time (s)	*2.943*	*2.818*	*3.154*
Overshoot (%)	*10.12*	*1.05*	*0.93*
Elapsed time (s)	***21009.3397***	***29053.3585***	***25266.4916***

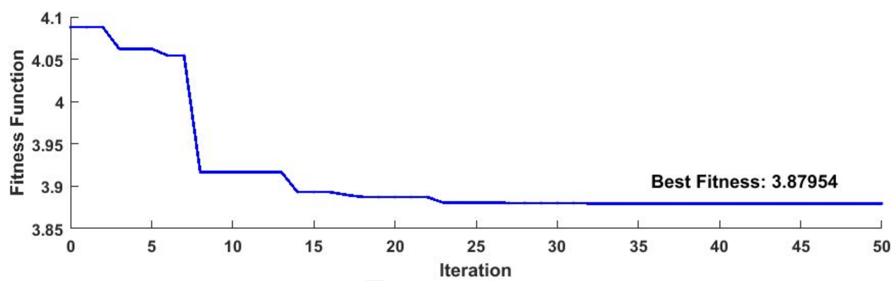


(a)

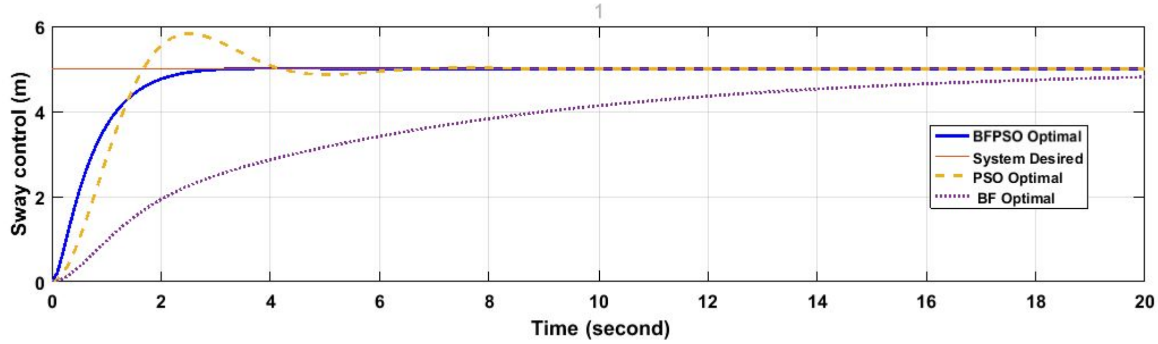


(b)

Figure 4. Hovercraft Surge control. (a) fBFPSO algorithm fitness value, (b) The proposed algorithm.

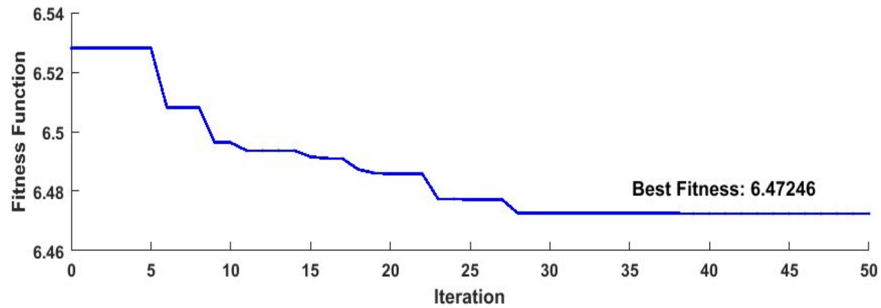


(a)

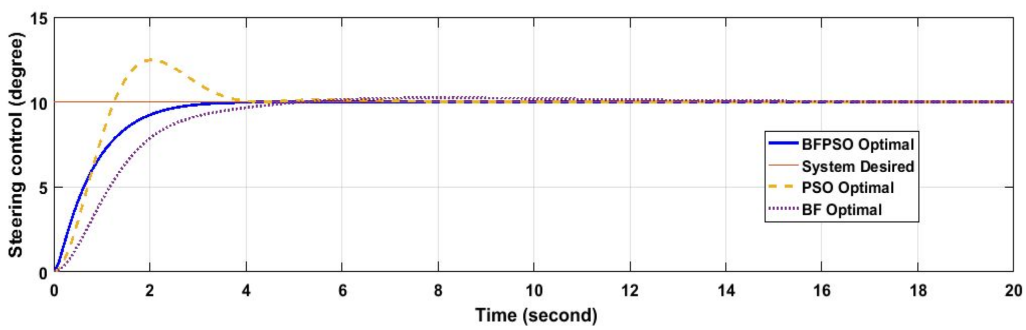


(b)

Figure 5. Hovercraft Sway control. (a) fBFPSO algorithm fitness value, (b) The proposed algorithm



(a)



(b)

Figure 6. Hovercraft Steering angle control. (a) fBFPSO algorithm fitness value, (b) The proposed algorithm.

5. Conclusions

The paper presents a successful design of three numerical optimal controllers to pilot an autonomous hovercraft. The fBF-PSO algorithm that employs both the ability to exchange social information of the PSO algorithm and the capability of finding new BF solutions via the elimination and dispersal step is proposed in order to find the optimal PD like Fuzzy controller parameters for the hovercraft pilot models: surge, sway and steering control. The proposed controller obtains optimal pilot performance after the third second when compared with the other single optimal controller designs, PSO or BF. Furthermore, it provides fastest response, high reliability as well as stability than available controllers for the realistic hovercraft mathematical models. A path planning algorithm for hovercraft model with obstacle avoidance and wind disturbances will be developed in future. In addition, the real hovercraft model with the proposed control technique will be implemented for testing and optimization.

References

1. Cabecinhas D, Batista P, Oliveira P and Silvestre C. "Hovercraft Control with Dynamic Parameters Identification," in IEEE Transactions on Control Systems Technology, vol. 26, no. 3, pp. 785-796, May 2018, doi: 10.1109/TCST.2017.2692733.
2. Hebertt Sira-Ramirez and Carlos Aguilar Ibanez. “On the control of the Hovercraft”. Dynamic and Control, Vol. 10, pp. 151-163, Kluwer Academic Publishers, 2000.
3. Silvano Balemi, Robert Bucher, Paola Guggiari, Ivan Furlan, Markus Kottmann and Jacques Chapuis. “Rapid control of prototyping of a Hovercraft”, Conference: Proc. of MSy'02, Embedded Systems Conference, October, 2002.

4. Rashid, Aras, Kassim, Ibrahim, Jamali. "Dynamic Mathematical Modeling and Simulation Study of Small-scale Autonomous Hovercraft", International Journal of Advanced Science and Technology, Vol. 46, September, 2012.
5. Dictino Chaos, David Moreno-Salinas, Rocío Muñoz-Mansilla, and Joaquín Aranda, "Nonlinear Control for Trajectory Tracking of a Nonholonomic RC-Hovercraft with Discrete Inputs," Mathematical Problems in Engineering, vol. 2013, Article ID 589267, 16 pages, 2013.
6. Garcia DI and White WN. "Control design of an unmanned hovercraft for agricultural applications," International Journal of Agricultural and Biological Engineering, vol. 8, no. 2, pp. 72–79, 2015.
7. Mamdani EH. "Application of fuzzy algorithms for control of simple dynamic plant," Proceedings of the Institution of Electrical Engineers, vol. 121, no. 12, pp. 1585–1588, 1974.
8. Mamdani EH. "Application of fuzzy logic to approximate reasoning using linguistic synthesis," IEEE Transactions on Computers, vol. C-26, no. 12, pp. 1182–1191, 1977.
9. Zhao ZY, Tomizuka M, and Isaka S. "Fuzzy gain scheduling of PID controllers," IEEE Transactions on Systems, Man and Cybernetics, vol. 23, no. 5, pp. 1392–1398, 1993.
10. Sanchez EN, Becerra HM and Velez CM. "Combining fuzzy, PID and regulation control for an autonomous mini-helicopter", Information Sciences 177, pp. 1999–2022, 2007.
11. Precup RE and Preitl S. "PI-Fuzzy controllers for integral plants to ensure robust stability", Information Sciences 177, pp. 4410–4429, 2007.
12. Juang YT, Chang YT and Huang CP. "Design of fuzzy PID controllers using modified triangular membership functions", Information Sciences 178, pp. 1325–1333, 2008.
13. Tran HK, Son HH, Duc PV, Trang TT and Nguyen HN. "Improved Genetic Algorithm Tuning Controller Design for Autonomous Hovercraft". Processes 2020, 8, 66. https://doi.org/10.3390/pr8010066.
14. Passino KM and Yurkovich S. Fuzzy Control, Addison-Wesley, Reading, MA. 1998.
15. Eiben AE and Smith JE. Introduction to Evolutionary Computing; Springer: Berlin, Germany, 2003.
16. Yang, Xin-She. Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley&Sons, Inc. University of Cambridge, 2010.
17. Kennedy J. and Eberhart RC. "Particle swarm optimization", Proceedings IEEE International Conference on Neural Networks IV pp. 1942–1948, 1995.
18. Clerc M. "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization", Proceedings of the ICEC, Washington, DC, pp. 1951–1957, 1999.
19. Eberhart RC and Shi Y. "Comparing inertia weights and constriction factors in particle swarm optimization", Proc. of the IEEE Congress on Evolutionary Computation, USA pp. 84–88, 2000.
20. Bergh F and Engelbrecht AP. "A Cooperative approach to particle swarm optimization," IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 225-239, 2004.
21. Huang T and Mohan AS. "A hybrid boundary condition for robust particle swarm optimization", IEEE Antennas and Wireless Propagation Letters 4, pp. 112–117, 2005.
22. Passino KM. "Biomimicry of bacterial foraging for distributed optimization and control," IEEE Control Syst. Mag. 22 (2002) 52–67.
23. Juang CF. "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design", IEEE System, Man, and Cybernetics: B 34 (2), pp. 997–1006, 2004.
24. Li LL, Wang L and Liu LL. "An effective hybrid PSOSA strategy for optimization and its application to parameter estimation", Applied Mathematics and Computation 179, pp. 135–146, 2006.
25. Ali AF and Tawhid MA. "A Hybrid PSO and DE Algorithm for Solving Engineering Optimization Problems", Appl. Math. Inf. Sci. 10, No. 2, pp. 431-449, 2016.

26. Chen CL, Lin YL, Feng YC. "Optimization of large-scale economic dispatch with valve-point effects using a modified hybrid PSO-DSM approach", Journal of Marine Science and Technology, 2018. DOI: 10.6119/JMST-015-1215-1
27. Hsu CI, Wu SPJ, Chiu CC. "A Hybrid Swarm Intelligence Approach for Blog Success Prediction", International Journal of Computational Intelligence Systems, Vol. 12, Iss. 2, pp. 571 – 579, 2019.
28. Liu XiaoLong, Li RongJun and YangPing. "A bacterial foraging global optimization algorithm based on the particle swarm optimization," 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, 2010, pp. 22-27, doi: 10.1109/ICICISYS.2010.5658828.
29. Gu Q, Yin K, Niu B, Chen H. "RFID Networks Planning Using BF-PSO". In: Huang, DS., Ma, J., Jo, KH., Gromiha, M.M. (eds) Intelligent Computing Theories and Applications. ICIC 2012. Lecture Notes in Computer Science(), vol 7390. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31576-3_24
30. Kora P, Kalva SR. "Hybrid Bacterial Foraging and Particle Swarm Optimization for detecting Bundle Branch Block". Springer Plus 4, 481 (2015). https://doi.org/10.1186/s40064-015-1240-z
31. Korani WM, Dorrah HT and Emara HM. "Bacterial foraging oriented by Particle Swarm Optimization strategy for PID tuning," 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA), 2009, pp. 445-450, doi: 10.1109/CIRA.2009.5423165.
32. Abd-Elazim SM, Ali ES. "A hybrid Particle Swarm Optimization and Bacterial Foraging for optimal Power System Stabilizers design," International Journal of Electrical Power & Energy Systems, Volume 46, 2013, Pages 334-341, https://doi.org/10.1016/j.ijepes.2012.10.047.
33. Ibrahim HEA, Hassan FN, Anas O. Shomer. "Optimal PID control of a brushless DC motor using PSO and BF techniques", Ain Shams Engineering Journal, Volume 5, Issue 2, 2014, Pages 391-398, ISSN 2090-4479, https://doi.org/10.1016/j.asej.2013.09.013
34. El-Wakeel, Amged & A.kamel, & Abdel-hamed, Alaa. "A Hybrid BF-PSO Optimization Technique for Optimal Tuning of PID Controller of a Permanent Magnet Brushless DC Motor". Electric Power Components and Systems. 43. 309-319, 2015. 10.1080/15325008.2014.981320.
35. Amit Kumar and Sathans, "Hybrid Bacterial Foraging Enhanced PSO Algorithm for Load Frequency Control of Hydro-Thermal Multi-Area Power System and Comparative analysis", International Journal of Computing and Digital Systems, Vol. 9, Iss. 1, ISSN (2210-142X), pp:129-138, 2020.
36. Kiran HU, Tiwari SK. "Hybrid BF-PSO Algorithm for Automatic Voltage Regulator System". In: Gupta, D., Khanna, A., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1166, 2021. Springer, Singapore. https://doi.org/10.1007/978-981-15-5148-2_13
37. Skogestad S. "Simple analytic rules for model reduction and PID controller tuning", Journal of Process Control, Vol. 13, pp. 291-309, 2003.
38. Martins FG. "Tuning PID Controllers using the ITAE Criterion" International Journal Engineering Education, 21(5), pp. 867-873, 2005.
39. Tan W, Liu J, Chen T, Marquez HJ. "Comparison of some well-known PID tuning formulas". Computation Chemical Engineering, pp. 1416–1423, 2006.