A Kalman-Filter-Based Sine-Cosine Algorithm

Mohd Falfazli Mat Jusof[1], Shuhairie Mohammad, Ahmad Azwan Abd Razak, Ahmad Nor Kasruddin Nasir[2], Mohd Riduwan Ghazali, Mohd Ashraf Ahmad, Addie Irawan Hashim
Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang
26600, Pekan, Pahang
[1]mfalfazli@ump.edu.my, [2]kasruddin@ump.edu.my

***Abstract*—This paper presents a Kalman-Filter-based Sine Cosine algorithm (KFSCA). It is a synergy of a Simulated Kalman Filter (SKF) algorithm and a Sine Cosine (SCA) algorithm. SKF is a random based optimization algorithm inspired from the Kalman Filter theory. A Kalman gain is formulated following the prediction, measurement and estimation steps of the Kalman filter design. The Kalman gain is utilized to introduce a dynamic step size of a search agent in the SKF algorithm. On the other hand, a Sine Cosine algorithm is formulated based on mathematical sine and cosine terms. A random based searching strategy is formulated through a little modification on both of the terms. In the KFSCA, a Kalman gain is introduced to vary an individual agent's step and thus balances exploration and exploitation strategies of the original SCA. Cost function value that represent an accuracy of a solution is considered as the ultimate goal. Every single agent carries an information about the accuracy of a solution in which will be used to compare with other solutions from other agents. A solution that has a lower cost function is considered as the best solution. The algorithm is tested with various benchmark functions and compared with the original SCA algorithm. Result of the analysis on the accuracy tested on the benchmark functions is tabulated in a table form and shows that the proposed algorithm outperforms SCA significantly. The result also is presented in a graphical form to have a clearer visual on the solution.**

Keywords*— *sine-cosine, optimization algorithm, Kalman Filter, varying step.

I. Introduction

Research on Soft Computing method is gaining attentions from researchers worldwide. It is an area of research that has a great combination of self-learning, intelligence, fuzziness, etc. Due to a rapid progressing of computing machine technology that could provide high processing speed, the research on Soft Computing is more promising. Branches of research in Soft Computing include Evolutionary Computation, Machine Learning, Fuzzy Logic and Swarm Intelligence. All these four branches are very much inter-related and complimentary to each other.

Incorporation of various intelligent features into a group of population is the basis of a Swarm Intelligence. One popular application of Swarm Intelligence is optimization algorithm. The Swarm Intelligence optimization algorithm is also known as multi-agents or population based optimization algorithm. It has been applied to solve many real world problems with high complexity and large dimension. More importantly, solution resulted from multi-agents based algorithm is more promising and reliable for a real life application. Multi-agents based optimization algorithm is divided into several categories. They include biological-inspired and non-biological inspired optimization algorithms in which their searching strategy is formulated based on living creature and non-living creature respectively. Example algorithms from bio-inspired category include Bacterial Foraging Algorithm [1] and Particle Swarm Algorithm [2] while example algorithms inspired from non-living creature are Spiral Dynamic Algorithm [3], Sine-Cosine algorithm [4] (SCA) and Simulated Kalman Filter (SKF) algorithm [5].

Kalman Filtering theory was developed by Kalman [6]. It has been used widely in engineering application for tracking and navigation as well as predicting and estimating for various physical systems. With the ability to predict and estimate, [7] applied the Kalman Filtering theory to develop population-based optimization algorithm named Simulated Kalman Filter (SKF). With the emergence of SKF optimizing algorithm, research on the Kalman Filter has expanded to a wider range of area and application. The research on SKF has grown through a development of some parts of its components. [8] introduced adaptive Kalman gain with respect to number of function evaluation. It was applied to solve a liquid slosh problem on a moving transport. [9] introduced a multiobjective SKF called MOSKF. It was tested with various multiobjective benchmark functions in comparison to Non-dominated Sorting Genetic Algorithm II (NSGAII). Both NSGAII and MOSKF had a competitive performance to each other.

SCA on the other hand, is formulated based on mathematical sine and cosine terms. Through a manipulation of the terms, a random based searching strategy can be generated. Also, adding an elitism component into the technique as a guide to every individual searching agent may result to a higher performance algorithm. A lot of modifications have been done since the introduction of SCA. [10] synergized a Crow Search Algorithm (CSA) with the SCA. An elitism strategy from the SCA was incorporated into the CSA. The proposed algorithm successfully increased accuracy when tested with various benchmark functions. [11] incorporated both sine and cosine update equations from SCA into water wave algorithm. An opposition-based strategy was also adopted to enhance exploration capability. The proposed algorithm was known as Sine Cosine Water Wave optimization (SCWWO). The proposed strategy successfully improved both speed and accuracy. [12] proposed a Chaos Cultural Sine Cosine Algorithm. Both sine cosine update equations from SCA and a chaotic strategy were adopted into Cultural algorithm. The algorithm was employed to optimize operation in a hydro-station.

Motivated by the aforementioned scenarios, this paper proposes a Kalman Filter-based Sine Cosine Algorithm (KFSCA). It is a synergy of a Kalman filter gain and a SCA. The rest of this paper is organized as follows. The first part of Section II explains the SKF and SCA algorithms. The second part of the Section II explains the proposed KFSCA. In Section III, some of the benchmark functions used to test the performance of the proposed algorithm and its corresponding test setup are presented. Section IV discusses the result and finding of the work. Finally, Section V presents the conclusion of the work presented in this paper.

II. Simulated Kalman Filter, Sine-Cosine Algorithm and Kalman-Filter-based Sine Cosine Algorithm

A. Simulated Kalman Filter

Simulated Kalman Filter (SKF) is governed from a theory of a physical system. It consists of three main phases. The first phase is called a Prediction phase. It is a phase where a future state of a physical system is predicted. The prediction proses is conducted based on the previous information of the physical system and is associated with a prediction noise and a variance in the Prediction phase. Equation in the Prediction phase is shown as (1).

$$P(t+1) = P(t) + Q \tag{1}$$

where $P(t)$ is a variance associated with the prediction at a current time, t and Q is a prediction noise covariance in the prediction. The value of $P(t)$ changes over time while the Q is a constant.

The second phase is a Measurement phase. It is a phase where an actual information about a particular state of the physical system is updated. It involves calculations of a measurement noise and a variance associated in the Measurement phase. The last phase is called an Estimation phase. It consists of an adaptive Kalman gain in which calculated based on estimation noise and a variance associated with the estimation. The Kalman gain is represented as (2)

$$K(t) = \frac{\widehat{P}(t)}{\widehat{P}(t) + R} \tag{2}$$

where $\widehat{P}(t)$ is a variance associated in the estimation at time t and R is an estimation noise covariance. The $\widehat{P}(t)$ and R are defined as a vector and a constant respectively. The $\widehat{P}(t)$ is always updated and can be represented as (3).

$$\widehat{P}(t) = (1 - K(t)) \times \widehat{P}(t) \tag{3}$$

B. Sine-Cosine Algorithm

Sine – cosine algorithm begins with an initialization of a position of search agents. The search agents are then moved based on update equations as represented in (4) and (5).

$$X_i^{t+1} = X_i^t + r_1 \times \sin(r_2) \times r_3 \left| P_i^t - X_i^t \right| \tag{4}$$

$$X_i^{t+1} = X_i^t + r_1 \times \cos(r_2) \times r_3 \left| P_i^t - X_i^t \right| \tag{5}$$

where r_1 is an adaptive step size of the search agent's, , r_2 is a random equation for the sine and cosine terms and r_3 is a random parameter of an individual agent's position relative to the best position in the agents population. The r_1 is defined as (6).

$$r_1(t) = a - t\frac{a}{T} \tag{6}$$

where a is a tunable constant, t is an iteration at a current time, and T is the maximum iteration. The equation r_1 is formulated such that it is decreased as the current iteration is increased. The equation r_2 is defined as (7).

$$r_2 = 2 \times \pi \times rand \tag{7}$$

where *rand* is a random number. It represents a stochastic strategy of the algorithm in deciding an agent's motion. The parameter r_3 is defined as a random number applied to each agent's position.

C. Kalman Filter-based Sine-Cosine Algorithm

A detailed-description of the Kalman Filter-based Sine-Cosine Algorithm (KFSCA) is shown in Figure 1. In the algorithm, Prediction and Estimation phases of the SKF are adopted into SCA. The Prediction is applied prior to update sine and cosine equations as in Step4. On the contrary, the Estimation is employed after the update position as in Step6. The Measurement phase of SKF is replaced by the update equations of the original SCA as in Step5.

Step1: Initialize i^{th} agents' position, X_i and maximum iteration, T.

Step2: Update cost function, $f(X_i)$ value of every single agent, i.

Step3: Determine the best agent X_{fmin}. Agent with lowest cost function value, f_{min} is considered as the best agent.

Step4: Apply Prediction equation of the Simulated Kalman filter as (1).

Step5: Update every single individual i_{th} agent's position using equations (4) and (5). This is equivalent to Measurement phase of the Simulated Kalman filter.

Step6: Apply Estimation equations of the Simulated Kalman filter as (2) and (3).

Step7: Update cost function, $f(X_i)$ value of every single agent, i.

Step8: Determine the best agent X_{fmin}. Agent with lowest cost function value, f_{min} is considered as the best agent.

Step9: Check if the iteration has reached the maximum value. If it is true, stop the algorithm. If it is not true, repeat Step4 until Step9.

Fig. 1. Description of the Kalman Filter-based Sine-Cosine algorithm.

III. BENCHMARK FUNCTIONS AND TEST SETUP

Performance of a newly formulated algorithm in terms of its accuracy can be tested using various benchmark functions. An algorithm is considered has the best accuracy solution and optimal performance if it can find the theoretical global optima point. Benchmark functions that have been adopted in this work are shown in Table I [1]. Theoretical global optima point for all functions in the table is at point zero and they are all characterized by shifted and unimodal. Functions 1, 3, 4 have a search range of [-100, 100]. On the other hand, functions 2 and 5 have a search range of [-10, 10] and [-30, 30] respectively.

Following the work of [4], the performance test was setup with 100 agents and maximum of 500 iterations. For the purpose of a statistical analysis, 30 repeated runs were conducted for both algorithms. Parameters *P*, *Q*, and *R* for the KFSCA were defined as 1000, 0.5 and 0.5 respectively and the maximum adaptive step size, *a* for both KSCA and SCA was defined as 2.

TABLE I. BENCHMARK FUNCTIONS

Func. No	Benchmark functions		
	Mathematical representation	***Range***	f_{min}
f_1	$\sum_{i=1}^{n} x_i^2$	± 100	*0*
f_2	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	± 10	*0*
f_3	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	± 100	*0*
f_4	$\max_i \{ \lvert x_i \rvert,\ 1 < i \le n \}$	± 100	*0*
f_5	$\sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$	± 30	*0*

IV. RESULT AND DISCUSSION

Result of the performance test for SCA, SKF and KFSCA is presented in both graphical and a numerical value. Based on the 30 repeated runs, an average value of the cost function value, the worst and the best cost function value are recorded. The smallest cost function value is considered as the best and it represents high accuracy attainment. On the contrary, the highest cost function value is considered as the worst and it represents low accuracy attainment.

Tables II, III and IV show accuracy attainment of SCA, SKF and KFSCA algorithms respectively. Winner of the average, the best and the worst is highlighted in bold font. Noted that KFSCA achieved the best for the mean and best accuracy performance followed by the SCA and SKF. For the worst accuracy achievement, the first place is attained by SKF followed by SCA and KFSCA.

Based on result presented in Tables II to IV, a statistical analysis was then conducted. Wilcoxon Sign Rank test was selected as the method of analysis. It is to check a significant difference between any two algorithms. This can be done by calculating and observing the *p*-value acquired from the statistical test. If the calculated *p*-value is lower than 5% or 0.05, the compared results are considered as significantly different to each other.

TABLE II. ACCURACY ATTAINMENT OF THE SCA

Func. No	Accuracy attainment result		
	Best	***Worst***	***Mean***
f_1	1.7061×10^{-5}	0.5501	0.0421
f_2	2.7917×10^{-6}	0.0028	3.0108×10^{-4}
f_3	2.8182	3.6113×10^{3}	9.7745×10^{2}
f_4	0.5203	32.1078	7.0976
f_5	18.6265	1.3686×10^{3}	1.5185×10^{2}

TABLE III. ACCURACY ATTAINMENT OF THE SKF

Func. No	Accuracy attainment result		
	Best	***Worst***	***Mean***
f_1	*27.9300*	**8.9076×10^{3}**	*2.0651×10^{3}*
f_2	*0.2093*	**24.0572**	*6.7007*
f_3	*3.4676×10^{3}*	**1.5861×10^{4}**	*1.0385×10^{4}*
f_4	*35.0983*	**75.6344**	*53.8395*
f_5	*1.6716×10^{5}*	**1.0817×10^{7}**	*3.7158×10^{6}*

TABLE IV. ACCURACY ATTAINMENT OF THE KFSCA

Func. No	Accuracy attainment result		
	Best	***Worst***	***Mean***
f_1	**7.6339×10^{-140}**	7.2876×10^{-79}	**2.5667×10^{-80}**
f_2	**3.1065×10^{-81}**	6.6286×10^{-42}	**4.3683×10^{-43}**
f_3	**1.3671×10^{-68}**	2.0735×10^{-49}	**8.6806×10^{-51}**
f_4	**3.5914×10^{-81}**	2.5595×10^{-39}	**9.7539×10^{-41}**
f_5	**0.0794**	18.9541	**18.2416**

Results of the Wilcoxon test for comparing KFSCA and SCA and for comparing KFSCA and SKF is shown in Table V. Noted that the test achieved *p*-value 1.73×10^{-6} which is less than 0.05 for all test functions. Other results of the test such as sum of positive and negative ranks and *z*-value are also shown in Table V. From these results, it can be summarized that KFSCA result for functions f_1 to f_5 is significantly different to the result of SCA and SKF.

TABLE V. WILCOXON SIGN RANK TEST

Func. No	Wilcoxon test result between SCA and LASCA and between SCA and EASCA			
	Sum of +ve rank	***Sum of -ve rank***	***z-value***	***p-value***
f_1 to f_5	465	0	-4.7821	1.73×10^{-6}

The result is tested with Friedman test. It is to show overall ranking of the algorithms. Friedman test was conducted based on the mean value presented in Tables II to IV. The Friedman test result that showing the ranking of each algorithm is shown in Table VI. Noted that, for all test functions, KFSCA achieved the first rank followed by the SCA and SKF. However, for function f_5, the performance of KFSCA is not good as its performance for functions f_1 to f_4.

TABLE VI. FRIEDMAN TEST RESULT

Func. No.	Rank		
	SCA	***SKF***	***KFSCA***
f_1 to f_4	2.0	*3.0*	**1.0**
f_5	1.9	*3.0*	**1.1**

Graphical representation of the result comparing performance of the algorithms tested on functions f_1 to f_5 are shown in Figures 2 to 6. KFSCA, SCA and SKF graphs are represented in blue dashed line, black dashed-dotted line and red smoothed line respectively. Noted that graphs of SCA show slower convergence compared to both SKF and KFSCA. However it achieved better accuracy compared to SKF. On the contrary, SKF achieved faster convergence compared to SCA, but its accuracy is the worst among the 3 algorithms. KFSCA on the other hand, show the best for both convergence speed and accuracy. The presented graphs telly with the result presented in Tables II to IV.

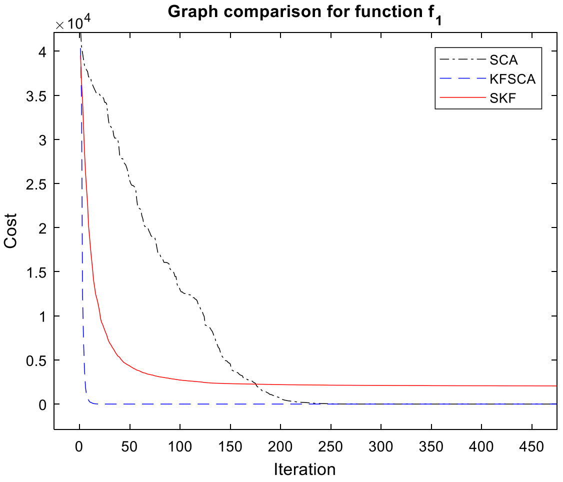

Fig. 2. Convergence plot for function f_1

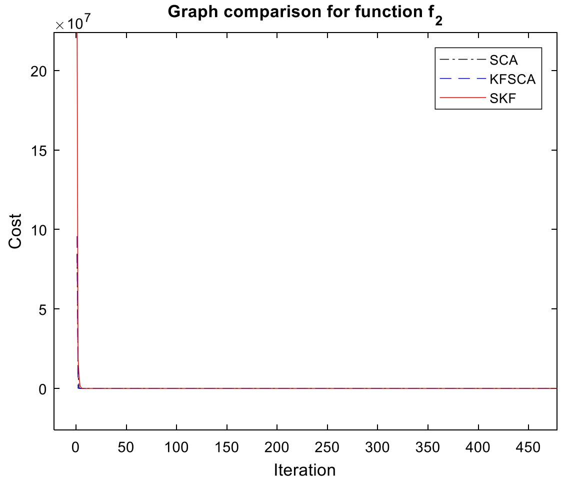

Fig. 3. Convergence plot for function f_2

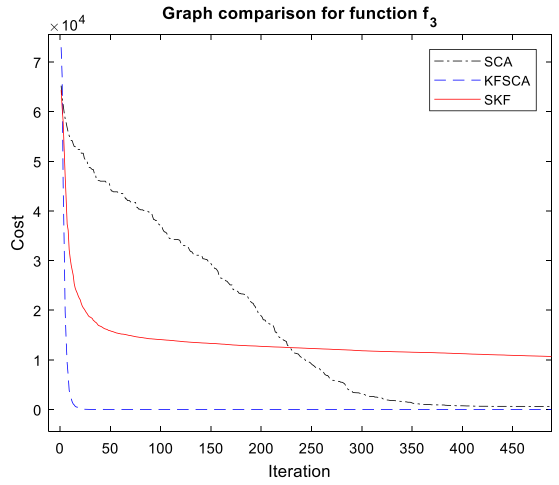

Fig. 4. Convergence plot for function f_3

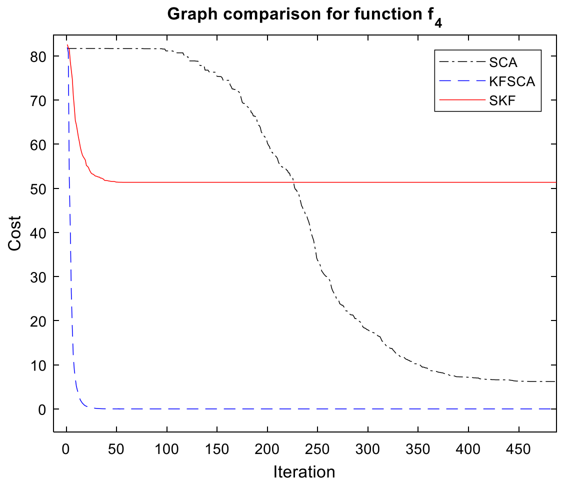

Fig. 5. Convergence plot for function f_4

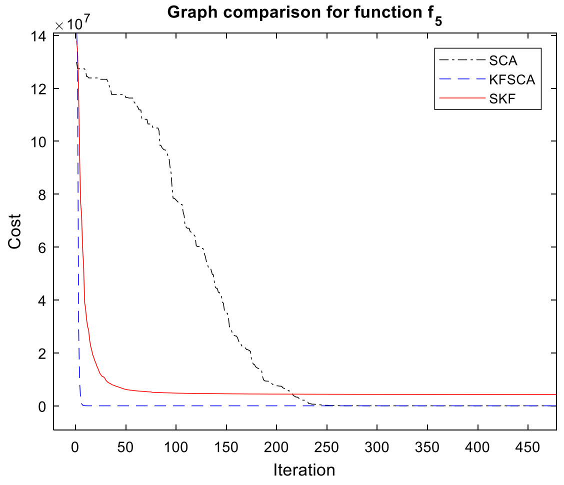

Fig. 6. Convergence plot for function f_5

V. CONCLUSION

A Kalman Filter Sine Cosine Algorithm is presented in this paper. It is a hybrid version, synergizing Kalman gain update strategy and Sine Cosine Algorithm (SCA). Prediction and Estimation phases of Simulated Kalman Filter (SKF) have been incorporated into SCA in which varies population step size. The algorithm has been tested on 5 different benchmark functions in comparison with SCA and SKF. Result of the test has been statistically analysed using non-parametric Wilcoxon Sign Rank test and Friedman test. Analysis of the test, has been shown that KFSCA significantly outperformed both SKF and SCA in both accuracy and convergence speed performances. The proposed algorithm will be further tested comprehensively using other state-of-the-art benchmark functions and will be applied to solve a complex and nonlinear problem in system engineering.

ACKNOWLEDGMENT

This research is fully funded by UMP research grant RDU1703149 where it is awarded by the Automotive Engineering Centre (AEC) and the Department of Research and Innovation through Faculty of Electrical and Electronics Engineering of the Universiti Malaysia Pahang (UMP).

REFERENCES

[1] Kevin M Passino (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems, 22 (3), pp. 52-67.

[2] Kennedy J. (2011) Particle Swarm Optimization. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[3] Tamura K. and Yasuda K. (2011). Primary study of spiral dynamics inspired optimization. IEEJ Transactions on Electrical and Electronic Engineering, 6 (S1), pp. S98–S100

[4] Seyedali Mirjalili (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-BasedSystems 000 (2016) pp. 1–14.

[5] Z. Ibrahim, N. H. A. Aziz, N. A. A. Aziz, S. Razali, M. S. Mohamad (2016). "Simulated Kalman Filter: A Novel Estimation-Based Metaheuristic Optimization Algorithm", Advance Science Letters, Vol. 22, pp. 2941-2946

[6] R. E. Kalman (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82 (1), pp. 35-45.

[7] Ibrahim, Z, Abdul Aziz N.H., Ab. Aziz N.A., Razali, S., Shapiai M.I., Nawawi S.W. and Mohamad M.S. (2015) A Kalman Filter Approach for Solving Unimodal Optimization Problems. ICIC Express Letters, 9 (12). pp. 3415-3422.

[8] Jusof MFM, Nasir ANK, Ahmad MA and Ibrahim Z (2018). An Exponential based Simu-lated Kalman Filter Algorithm for Data-driven PID Tuning in Liquid Slosh Controller. Pro-ceeding of 2018 IEEE International Conference on Applied System Invention (ICASI), 13-17 April 2018, Chiba Japan, pp. 984-987

[9] A. Azwan A. Razak, Jusof MFM, Nasir ANK and Ahmad MA (2018). A Multiobjective Simulated Kalman Filter Optimization Algorithm, Proceeding of 2018 IEEE International Conference on Applied System Invention (ICASI), 13-17 April 2018, Chiba Japan, pp. 23-26.

[10] Pasandideh SHR, Khalilpourazari S (2018) Sine cosine crow search algorithm: A powerful hybrid metaheuristic for global optimization. Cornell Univer Library 1–15. arXiv :1801.08485.

[11] Zhang J., Zhoua Y. and Luo Q. (2018).An improved sine cosine water wave optimization algorithm for global optimization, Journal of Intelligent & Fuzzy Systems 34 (2018) 2129–2141.

[12] Zou Q., Li A., He X. and Wang X. (2018). Optimal Operation of Cascade Hydropower Stations Based on Chaos Cultural Sine Cosine Algorithm. IOP Conf. Series: Materials Science and Engineering 366 (2018).012005.