mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 1 – 7
DOI: https://doi.org/10.15282/mekatronika.v1i2.4794

ORIGINAL ARTICLE

Solving Assembly Sequence Planning Using Distance Evaluated Simulated Kalman Filter

Ainizar Mustafa[1], Zuwairie Ibrahim[2], Zulkifli Md Yusof[2], David Al-Dabass[3] and Philip Sallis[4]

[1]Isuzu Hicom (M) Sdn Bhd, Kawasan Perindustrian Peramu Jaya, 26607 Pekan, Pahang, Malaysia.
[2]Faculty of Mechanical and Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.
[3]Nottingham Trent University, United Kingdom.
[4]School of Engineering, Computer and Mathematical Sciences, Aukland University of Technology, New Zealand.

ABSTRACT – This paper presents an implementation of simulated Kalman filter (SKF) algorithm for optimizing an assembly sequence planning (ASP) problem. The SKF search strategy contains three simple steps; predict-measure-estimate. The main objective of the ASP is to determine the sequence of component installation to shorten assembly time or save assembly costs. Initially, permutation sequence is generated to represent each agent. Each agent is then subjected to a precedence matrix constraint to produce feasible assembly sequence. In this paper, the distance evaluated SKF (DESKF) is proposed for solving ASP problem. The performance of the proposed DESKF is compared against previous works in solving ASP by applying BGSA, BPSO, and MSPSO. Using a case study of ASP, the results show that DESKF outperformed all the algorithms in obtaining the best solution.

ARTICLE HISTORY
Received: 28 February 2019
Accepted: 24 July 2019

KEYWORDS
Assembly Sequence Planning
Simulated Kalman Filter
Distance Evaluated

Introduction

In 2015, a new metaheuristic algorithm called simulated Kalman filter (SKF) has been proposed for numerical optimization problems [1-3]. The SKF operates using Kalman filtering process to solve optimization problems. After that many studies on SKF have been reported. For example, the SKF has been studied fundamentally [4-5]. The SKF also has been extended for combinatorial optimization problems [6-9]. Hybridization of SKF with particle swarm optimization (PSO), gravitational search algorithm (GSA), and opposition-based learning [10-15] have also been proposed for better performance. Other variants called parameter-less SKF and randomized SKF algorithms were proposed in [16-17]. The SKF has also been applied for real world problems like the adaptive beamforming in wireless cellular communication [18-21], airport gate allocation problem [22-23], feature selection of EEG signal [24-25], system identification [26-27], image processing [28-29], controller tuning [30], and printed circuit board (PCB) drill path optimization [31-32].

Assembly optimization in the production planning stage deals with determination of optimum assembly sequence and determination of optimum location of each resource. Solving the assembly sequence planning (ASP) problem is crucial because it will determine many assembly aspects including tool changes, fixture design and assembly freedom. Assembly sequence also influences overall productivity because it determines how fast and accurate the product is assembled.

Assembly sequence planning consists of assembly, operations, existing assembly techniques, and some details of relations between parts. Some researchers have dedicated their work on some important issues related to concurrent engineering analyses on assembly sequence planning. These issues are the representation of a product to be assembled, the generation of assembly sequence plans and the determination of precedence constraints, the representation of resulting assembly sequence plans, and the selection of the optimum assembly sequence planning. In this paper, assumptions for ASP are as follows: (1) setup time and the actual assembly time for each part and component are given, (2) transfer time between workstations is included in set up time, and (3) downtime of machines and workstations is omitted.

The total assembly time is the combination of setup time and actual assembly time. It is assumed that regardless of the assembly sequence, the actual assembly time is constant. A proper tool and setup for each component to be assembled is required. These two items depend on the geometry of the component itself and the components assembled to that point. The setup time for a component can be predicted using the following function:

$$Time_{Setup}(a) = p_{a0} + \sum_{b=1}^{c} p_{ab} q_{ab} \tag{1}$$

where a is the number of component to be assembled, is the setup time for product (a) being the first component, is the contribution to the setup time due to the presence of part (b) when entering part a, and if component (b) has already been assembled. Otherwise, Total assembly time is the summation of setup time and actual assembly time. Hence, the objective function for minimizing the assembly time is as follows:

$$\text{Min } Time_{Assembly} = \sum_{b=1}^{c} \left(Time_{Setup}(a) + A_a \right) \tag{2}$$

where A_a is the assembly time for component a.

Previously, the authors have solved the ASP using angle-modulated SKF, which is a variant of SKF algorithm established specifically for combinatorial optimization problems [33]. In this paper, the ASP is solved using another variant of SKF called distance evaluated SKF [8].

Distance Evaluated Simulated Kalman Filter (DESKF) Algorithm

The simulated Kalman filter (SKF) algorithm [1] is illustrated in Figure 1. Consider n number of agents, SKF algorithm begins with initialization of n agents, in which the states of each agent are given randomly. The maximum number of iterations, tmax, is defined. The initial value of error covariance estimate, P(0), the process noise value, Q, and the measurement noise value, R, which are required in Kalman filtering, are also defined during initialization stage.

Then, every agent is subjected to fitness evaluation to produce initial solutions $\{\boldsymbol{X}_1(0), \boldsymbol{X}_2(0), \boldsymbol{X}_3(0), \ldots, \boldsymbol{X}_{n-2}(0), \boldsymbol{X}_{n-1}(0), \boldsymbol{X}_n(0)\}$. The fitness values are compared and the agent having the best fitness value at every iteration, t, is registered as $\boldsymbol{X}_{\mathbf{best}}(t)$.

The-best-so-far solution in SKF is named as $\boldsymbol{X}_{\mathbf{true}}$. The $\boldsymbol{X}_{\mathbf{true}}$ is updated only if the $\boldsymbol{X}_{\mathbf{best}}(t)$ is better. In the prediction step, the following time-update equations are computed:

$$\boldsymbol{X}_i(t|t+1) = \boldsymbol{X}_i(t) \tag{3}$$

$$\boldsymbol{P}(t|t) = \boldsymbol{P}(t) + \boldsymbol{Q} \tag{4}$$

where where $\boldsymbol{X}_i(t)$ and $\boldsymbol{X}_i(t|t)$ are the current state and current transition/predicted state, respectively, and $\boldsymbol{P}(t)$ and $\boldsymbol{P}(t|t)$ are the current error covariance estimate and current transition error covariance estimate, respectively. Note that the error covariance estimate is influenced by the process noise, Q.

The next step is measurement, which is a feedback to estimation process. Measurement is modelled such that its output may take any value from the predicted state estimate, $\boldsymbol{X}_i(t|t)$, to the true value, $\boldsymbol{X}_{\mathbf{true}}$. Measurement, $\boldsymbol{Z}_i(t)$, of each individual agent is simulated based on the following equation:

$$\boldsymbol{Z}_i(t) = \boldsymbol{X}_i(t|t) + \sin(2\pi r_i(t)) \times |\boldsymbol{X}_i(t|t) - \boldsymbol{X}_{\text{true}}| \tag{5}$$

The $\sin(2\pi r_i(t))$ term provides the stochastic aspect of SKF algorithm and $r_i(t)$ is a uniformly distributed random number in the range of [0,1]. The final step is estimate. During this step, Kalman gain, $K(t)$, is computed as follows:

$$K(t) = \boldsymbol{P}(t|t)/(\boldsymbol{P}(t|t)+R) \tag{6}$$

Then, the estimation of next state, $\boldsymbol{X}_i(t+1)$, and the updated error covariant, $\boldsymbol{P}(t+1)$, are computed based on (7) and (8), respectively:

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t|t) + K(t) \times \delta \tag{7}$$

$$\boldsymbol{P}(t+1) = (1-K(t)) \times \boldsymbol{P}(t|t) \tag{8}$$

where $\delta = (\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t))$. Finally, the next iteration is executed until the maximum number of iterations, $t_{\max}$, is reached.

The distance evaluated simulated Kalman filter (DESKF) algorithm [8] is an extension of simulated Kalman filter (SKF) algorithm. The main idea of the distance evaluated approach in solving combinatorial optimization problem is to map the distance into a probabilistic value [0,1] and then the probabilistic value will be compared with a random number [0,1] to update a bit string or solution to a combinatorial optimization problem.

During the initialization of agents, the states of each agent are given randomly. In addition, every agent is associated with a random bit string as well. The length of the bit string is problem dependent and subjected to the size of the problem. Thus, 2 types of variables are associated with an agent in SKF. They are continuous variable, x, which is produced as estimated value of SKF (also similar to the position of agents in a search space), and a bit string, Σ, which is used to represent solution to a combinatorial optimization problem.

In DESKF, for a particular dth dimension, the distance between an ith agent to the best-so-far solution at iteration t, $\boldsymbol{D}_i(d, t)$, can be calculated as follows:

$$\boldsymbol{D}_i(d, t) = \boldsymbol{x}_i(d, t) - \boldsymbol{x}_{best\text{-}so\text{-}far}(d, t) \quad (9)$$

In DESKF, a probability function is used to map a velocity value into a probabilistic value within interval [0,1]. This distance value, $\boldsymbol{D}_i(d, t)$, is mapped to a probabilistic value within interval [0,1] using a probability function, $S(\boldsymbol{D}_i(d, t))$ as follows:

$$S(\boldsymbol{D}_i(d, t)) = |\tanh(\boldsymbol{D}_i(d, t))| \quad (10)$$

After the $S(\boldsymbol{D}_i(d, t))$ is calculated, a random number, *rand*, is generated and a binary value at dimension d of an ith agent, $\boldsymbol{\Sigma}_i(d, t)$, is updated according to the following rule:

if rand < $S(\boldsymbol{D}_i(d, t))$
  then $\boldsymbol{\Sigma}_i(t+1)$ = complement $\boldsymbol{\Sigma}_i(d, t)$
  else $\boldsymbol{\Sigma}_i(t+1) = \boldsymbol{\Sigma}_i(d, t)$
end

Distance Evaluated Simulated Kalman Filter (DESKF) For The ASP

A solution to an ASP is represented by a string of binary number. For example, if there are four components, binary code for each component is shown in Table 1. An assembly sequence 1-2-4-3 can be represented by 00011110. In this example, since four components are involved, only two bits binary number is needed. More bits will be required if the number of components larger.

In this study, the assembly of a hypothetical product consisting 19 components, which is taken from [33], is considered. Relationship between 19 components is illustrated in Figure 2. The relationship can also be translated into a precedence matrix (PM) and coefficient values as shown in Table 2 and Table 3. In this diagram, the components that are free to be assembled are the components that can be placed regardless of any part of a sequence. To find an optimal solution, each agent representing feasible assembly sequence must be evaluated to obtain its fitness value. The evaluation of the fitness value and feasibility test are done with referring to the PM. It is worth pointing out that the components of free to be assembled are the components that can be placed regardless of any part of a sequence.

As a result, each agent produces a feasible assembly sequence. The optimum one is then selected from the feasible assembly sequences by evaluating fitness of each agent. After the stopping condition is met, the performance of the DESKF can be investigated.

Experiments, Result and Discussion

The performance of DESKF is compared against some related metaheuristic methods such as Binary Gravitational Search Algorithm (BGSA) [34], Binary Particle Swarm Optimization (BPSO) [35], and Multi-State Particle Swarm Optimization (MSPSO) [36]. The parameters and its value used for DESKF, BGSA, BPSO, and MSPSO are presented in Table 4. The BPSO used a constant inertia weight, $\omega = 0.6$. On the other hand, the MSPSO used a linearly decreasing inertia weight which begins at $\omega = 0.9$ and decreases at $\omega = 0.4$. The quality of results of DESKF is then measured based on the fitness values of the best solutions in minimizing the total assembly time.

To simplify the understanding of this work, fitness or objective value and solution is now called total assembly time and feasible assembly sequence, respectively. The average (mean), minimum (min), and maximum (max) of total assembly time of 50 feasible assembly sequences, and the standard deviation (STD) are recorded. Table 5 presents comparison of the result of DESKF against BGSA, BPSO, and MSPSO. Based on the results given in Table 4, DESKF outperformed BGSA, BPSO, and MSPSO in minimizing total assembly time and obtaining minimum average time of the ASP problem. The minimum total assembly time obtained by DESKF is 503.80 unit of time with associated sequence of components suggested by the DESKF is 1-2-4-3-9-12-13-5-16-15-18-11-6-7-8-14-10-17-19.

The average assembly time of DESKF is 518.91 unit of time and this average value is better than BGSA, BPSO, and MSPSO, which indicate DESKF's consistency over 50 runs. The best sequences obtained by BGSA, BPSO, and MSPSO, are 2-1-4-9-3-12-5-13-15-18-16-6-11-7-8-10-14-17-19 (508.20 unit of time), 13-23-5-12-15-16-41-11-9-18-6-7-8-10-14-17-19 (515.80 unit of time), 2-4-3-1-9-12-5-13-15-18-16-11-6-7-8-10-14-17-19 (514.00 unit of time).

It is true that the finding reported in this paper is much dependent on the parameter values used by other algorithms, which are BGSA, MSPSO, and BPSO. However, in this study, there was no attempt to replicate the experiments of BGSA, MSPSO, and BPSO. All the results were taken from the published paper.

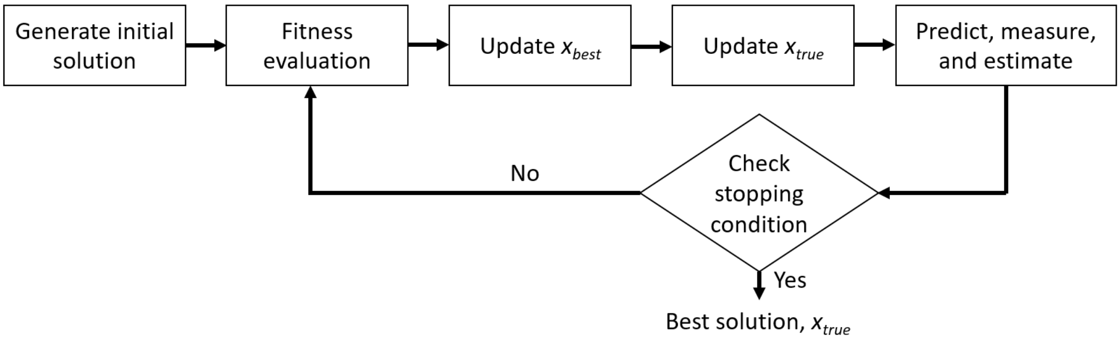

Figure 1. The simulated Kalman filter (SKF) algorithm.

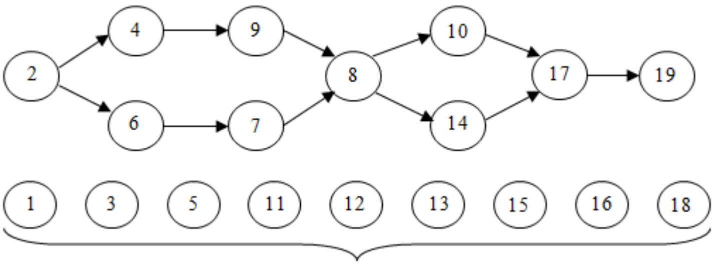

Figure 2. The assembly precedence diagram for the case study.

Table 1. Example of components number and its binary code.

Component number	Binary code
1	00
2	01
3	10
4	11

Table 2. Precedence matrix (PM) for the case study.

Component (a)	Component (b) 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1																			
2																			
3																			
4		1																	
5																			
6		1																	
7		1				1													
8		1		1		1	1		1										
9		1		1															
10		1		1		1	1	1	1										
11																			
12																			
13																			
14		1		1		1	1	1	1										
15																			
16																			
17		1		1		1	1	1	1	1				1					
18																			
19		1		1		1	1	1	1	1				1			1		

*CORRESPONDING AUTHOR | Ainizar Mustafa | ainizar.mustafa@isuzu-hicom.com.my

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 1 – 7
DOI: https://doi.org/10.15282/mekatronika.v1i2.2227

Table 3. Coefficient between components.

Comp. (a)	Comp. (b) 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	10	1	2	3	4	5	6	7	8	9	3.2	4.3	7	6.1	1.2	3.4	0	0	7.4
2	1.5	10	2	2	2	2	2	2	2	2	0	3.1	6	4.3	2.7	4.8	0	3	0.5
3	1	2.3	10	0	4	5	0	4	2.3	4.3	9.8	2.4	5	1.2	3.4	4.5	5.6	3.4	3.1
4	0	2	3.4	10	4.5	0	4	0	8	0	3.4	5.6	5	0	0	3.4	0	0	9.8
5	1.2	1	2	3	10	7.9	8.9	0	1.2	2	2.3	0	3	0	3.6	0	2.8	9.8	0
6	9.8	4.5	0	1.2	3.6	10	3.4	4	0	2.3	4.6	5.6	0	4	3	2	0	0.4	3.2
7	0.5	1.4	2.3	0.5	1.9	1	10	13.4	1.2	4	2.3	0	3	5.7	8.3	2	0.1	0	0.5
8	0	0	0	0	0	1.8	9.8	10	2.3	3	8.9	2.3	0	0	2.3	0.5	9.8	0	2.3
9	1	3	4.5	2.3	4.6	9.8	7.5	6.8	10	6	2.3	3.4	5	12.3	3.4	5.61	1	0	0
10	2.3	4.5	2.3	0	2.3	0	2.1	0	4.5	10	1.1	2.2	2	0	0	2.1	1.2	5.4	9.2
11	1	1	2	3	4	5	6	7	8	9	10	4.5	3	6.1	1.2	3.4	0.3	0	1.3
12	1.5	0	2	2	2	2	2	1	2	2	11.2	10	6	4.3	2.7	4.8	0	3	0.5
13	1	2.3	0	0	4	5	0	4	2.3	4.3	9.8	2.4	10	1.2	2.4	4.5	1.6	2.4	3.1
14	0	2	3.4	0	4.5	0	4	0	8	0	3.4	5.6	5	10	2.1	1.4	1	0	2.8
15	1.2	1	2	3	0	7.9	8.9	0	1.2	2	1.3	4	3	1.4	10	1.3	9.8	9.8	2
16	9.8	4.5	0	1.2	3.6	0	3.4	4	0	2.3	4.6	3.6	0	4	3	10	1.5	0	3.2
17	1	3	4	5	0	5	4	3.4	1.2	4	1.3	0	2	3.7	4.3	2.3	10	3.8	10
18	0.6	0.5	3.4	1.2	3	2	9.8	2	2.3	3	5.9	2.3	0	1	2.3	0.5	9.8	10	2.3
19	1	3	4.5	2.3	4.6	9.8	7.5	6.8	0	6	3.3	3	2	3.3	4.4	2.6	0.3	2.5	10

Table 4. Experimental setting.

Parameter	DESKF	BGSA	BPSO	MSPSO
Iteration	5000	500	500	500
Number of agents	10	50	50	25
Initial error covariance estimate, $P(0)$	100	-	-	-
Process noise, Q	0.5	-	-	-
Measurement noise, R	0.5	-	-	-
Inertia weight, ω	-	-	0.6	-
ω initial	-	-	-	0.9
ω final	-	-	-	0.4
Coefficient factor, c_1 and c_2	-	-	1.42	2

*CORRESPONDING AUTHOR | Ainizar Mustafa | ainizar.mustafa@isuzu-hicom.com.my

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 1 – 7
DOI: https://doi.org/10.15282/mekatronika.v1i2.2227

Conclusions

The ASP is a combinatorial optimization problem with a large-scale candidate solution. In this study, an approach based on a variant of SKF called DESKF is proposed to solve ASP problem. To evaluate the performance of the proposed approach, a case study of ASP consisting nineteen components is chosen, and the performance of DESKF is evaluated against three different approaches that uses BGSA, BPSO, and MSPSO as the search engine. Experimental results obtained showed that the proposed DESKF outperformed the other three approaches.

In future, the DESKF could be applied to solve ASP problem with different constraints such as assembly stability, machine and workstation assignment, and work load. Perhaps the experiments reported in this paper can be re-implemented with adjusted parameter values to get better result and more convincing comparison.

Acknowledgement

This research is supported by the Fundamental Research Grant Scheme (FRGS) awarded by the Ministry of Higher Education Malaysia to Universiti Malaysia Pahang (FRGS/1/2018/TK04/UMP/02/9).

References

[1] Ibrahim, Z., Abdul Aziz, N.H., Ab. Aziz, N.A., Razali, R. and Mohamad, M.S. (2016). Simulated Kalman filter: a novel estimation-based metaheuristic optimization algorithm. Advanced Science Letters, vol. 22, pp. 2941-2946.

[2] Abd Aziz, N.H., Ibrahim, Z., Razali, S. and Ab. Aziz, N.A. (2016) Estimation-based metaheuristics: a new branch of computational intelligence. The National Conference for Postgraduate Research, pp. 469-476.

[3] Abd Aziz, N.H., Ibrahim, Z., Razali, S., Bakare, T.A. and Ab. Aziz, N.A. (2016). How important the error covariance in simulated Kalman filter?. The National Conference for Postgraduate Research, pp. 315-320.

[4] Abd Aziz, N.H., Ab. Aziz, N.A., Mat Jusof, M.F., Razali, S., Ibrahim, Z., Adam, A. and Shapiai, M.I. (2018). An analysis on the number of agents towards the performance of the simulated Kalman filter optimizer. 8th International Conference on Intelligent Systems, Modelling and Simulation, pp. 16-21.

[5] Md Yusof, Z., Ibrahim, I., Satiman, S.N., Ibrahim, Z., Abd Aziz, N.H. and Ab. Aziz, N.A. (2015). BSKF: binary simulated Kalman filter. Third International Conference on Artificial Intelligence, Modelling and Simulation, pp. 77-81.

[6] Md Yusof, Z., Ibrahim, I., Ibrahim, Z., Abas, K.H., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2016). Local optimum distance evaluated simulated Kalman filter for combinatorial optimization problems. The National Conference for Postgraduate Research, pp. 892-901.

[7] Md Yusof, Z., Ibrahim, I., Ibrahim, Z., Mohd Azmi, K.Z., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2016). Distance evaluated simulated Kalman filter for combinatorial optimization problems. ARPN Journal of Engineering and Applied Sciences, vol. 11, pp. 4904-4910.

[8] Md Yusof, Z., Ibrahim, I., Ibrahim, Z., Mohd Azmi, K.Z., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2016). Angle modulated simulated Kalman filter algorithm for combinatorial optimization problems. ARPN Journal of Engineering and Applied Sciences, vol. 11, pp. 4854-4859.

[9] Muhammad, B., Ibrahim, Z., Mat Jusof, M.F., Ab. Aziz, N.A., Abd Aziz, N.H. and Mokhtar, N. (2017). A hybrid simulated Kalman filter - gravitational search algorithm (SKF-GSA). International Conference on Artificial Life and Robotics, pp. 707–710.

[10] Muhammad, B., Ibrahim, I., Mohd Azmi, K.Z., Abas, K.H., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2016). Performance evaluation of hybrid SKF algorithms: hybrid SKF-PSO and hybrid SKF-GSA. The National Conference for Postgraduate Research, pp. 865-874.

[11] Muhammad, B., Ibrahim, Z., Mohd Azmi, K.Z., Abas, K.H., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2016). Four different methods to hybrid simulated Kalman filter (SKF) with particle swarm optimization (PSO), The National Conference for Postgraduate Research, pp. 843-853.

[12] Muhammad, B., Ibrahim, Z., Mohd Azmi, K.Z., Abas, K.H., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2016). Four different methods to hybrid simulated Kalman filter (SKF) with gravitational search algorithm (GSA). The National Conference for Postgraduate Research, pp. 854-864.

[13] Muhammad, B., Ibrahim, Z., Ghazali, K.H., Mohd Azmi, K.Z., Ab. Aziz, N.A., Abd Aziz, N.H. and Mohamad, M.S. (2015). A new hybrid simulated Kalman filter and particle swarm optimization for continuous numerical optimization problems. ARPN Journal of Engineering and Applied Sciences, vol. 10, pp. 17171-17176.

[14] Ibrahim, Z., Mohd Azmi, K.Z., Ab. Aziz, N.A., Abd Aziz, N.H., Muhammad, B., Mat Jusof, M.F. and Shapiai, M.I. (2018). An oppositional learning prediction operator for simulated Kalman filter. The 3rd International Conference on Computational Intelligence and Applications, pp. 139-143

[15] Abd Aziz, N.H., Ibrahim, Z., Ab. Aziz, N.A. and Razali, S. (2017). Parameter-less simulated Kalman filter. International Journal of Software Engineering and Computer Systems, vol. 3, pp. 129-137.

[16] Abd Aziz, N.H., Ab. Aziz, N.A., Ibrahim, Z., Razali, S., Mat Jusof, M.F., Abas, K.H., Mohamad, M.S. and Mokhtar, N. (2017). Simulated Kalman filter with randomized Q and R parameters. International Conference on Artificial Life and Robotics, pp. 711-714.

[17] Lazarus, K., Noordin, N.H., Mat Jusof, M.F., Ibrahim, Z. and Abas, K.H. (2017). Adaptive beamforming algorithm based on a simulated Kalman filter. International Journal of Simulation: Systems, Science and Technology, vol. 18, pp. 10.1-10.5.

[18] Lazarus, K., Noordin, N.H., Mohd Azmi, K.Z., Abd Aziz, K.Z. and Ibrahim, Z. (2016). Adaptive beamforming algorithm based on generalized opposition-based simulated Kalman filter. The National Conference for Postgraduate Research, pp. 1-9.

*CORRESPONDING AUTHOR | Ainizar Mustafa | ainizar.mustafa@isuzu-hicom.com.my

[19] Lazarus, K., Noordin, N.H., Ibrahim, Z., Mat Jusof, M.F., Mohd Faudzi, M.A., Subari, N. and Mohd Azmi, K.Z. (2017). An opposition-based simulated Kalman filter algorithm for adaptive beamforming. IEEE International Conference on Applied System Innovation, pp. 91-94.

[20] Lazarus, K., Noordin, N.H., Ibrahim, Z. and Abas, K.H. (2016). Adaptive beamforming algorithm based on simulated Kalman filter. Asia Multi Conference on Modelling and Simulation, pp. 19-23.

[21] Md Yusof, Z., Satiman, S.N., Mohd Azmi, K.Z., Muhammad, B., Razali, S., Ibrahim, Z., Aspar, Z. and Ismail, S. (2015). Solving airport gate allocation problem using simulated Kalman filter. International Conference on Knowledge Transfer, pp. 121-127.

[22] Mohd Azmi, K.Z., Md Yusof, Z., Satiman, S.N., Muhammad, B., Razali, S., Ibrahim, Z., Ab. Aziz, N.A. and Abd Aziz, N.H. (2016). Solving airport gate allocation problem using angle modulated simulated Kalman filter. The National Conference for Postgraduate Research, pp. 875-885.

[23] Muhammad, B., Mat Jusof, M.F., Shapiai, M.I., Adam, A., Md Yusof, Z., Mohd Azmi, K.Z., Abdul Aziz, N.H., Ibrahim, Z. and Mokhtar, N. (2018). Feature selection using binary simulated Kalman filter for peak classification of EEG signals. 2018 8th International Conference on Intelligent Systems, Modelling and Simulation, pp. 1-6.

[24] Adam, A., Ibrahim, Z., Mokhtar, N., Shapiai, M.I., Mubin, M. and Saad, I. (2016). Feature selection using angle modulated simulated Kalman filter for peak classification of EEG signals. SpringerPlus, vol. 5, 1580.

[25] Mohd Azmi, K.Z., Ibrahim, Z., Pebrianti, D., Mat Jusof, M.F., Abdul Aziz, N.H. and Ab. Aziz, N.A. (2019). Enhancing simulated Kalman filter algorithm using current optimum opposition-based learning. Mekatronika, vol. 1, pp. 1-13.

[26] Mohd Azmi, K.Z., Ibrahim, Z., Pebrianti, D. and Mohamad, M.S. (2017). Simultaneous computation of model order and parameter estimation for ARX model based on single and multi swarm simulated Kalman filter. Journal of Telecommunication, Electronic, and Computer Engineering, vol. 9, pp. 151-155.

[27] Ann, N.Q., Pebrianti, D., Bayuaji, L., Daud, M.R., Samad, R., Ibrahim, Z., Hamid, R. and Syafrullah, M. (2018). SKF-based image template matching for distance measurement by using stereo vision. Intelligent Manufacturing and Mechatronics, pp. 439-447.

[28] Ann, N.Q., Pebrianti, D., Ibrahim, Z., Mat Jusof, M.F., Bayuaji, L. and Abdullah, N.R.H. (2018). Illumination-invariant image matching based on simulated Kalman filter (SKF). Journal of Telecommunication, Electronics and Computer Engineering, vol. 10, pp. 31-36.

[29] Muhammad, B., Pebrianti, D., Abdul Ghani, N., Abd Aziz, N.H., Ab. Aziz, N.A., Mohamad, M.S., Shapiai, M.I. and Ibrahim, Z. (2018). An application of simulated Kalman filter optimization algorithm for parameter tuning in proportional-integral-derivative controllers for automatic voltage regulator system. SICE International Symposium on Control Systems 2018, pp. 113-120.

[30] Abd Aziz, N.H., Ab. Aziz, N.A, Ibrahim, Z., Razali, S., Abas, K.H. and Mohamad, M.S (2016). A Kalman filter approach to PCB drill path optimization problem. IEEE Conference on Systems, Process and Control, pp. 33-36.

[31] Abd Aziz, N.H., Ibrahim, Z., Ab. Aziz, N.A., Md Yusof, Z. and Mohamad, M.S. (2018). Single-solution simulated Kalman filter algorithm for routing in printed circuit board drilling process. Intelligent Manufacturing and Mechatronics, pp. 649-655.

[32] Choi, Y.K., Lee, D.M. and Cho, Y.B. (2008). An approach to multi-criteria assembly sequence planning using genetic algorithms. International Journal of Advanced Manufacturing Technology, vol. 42, issue 1-2, pp. 180-188.

[33] Ibrahim, I., Ibrahim, Z., Ahmad, H., Md Yusof, Z., Shapiai, M.I., Nawawi, S.W. and Mubin, M. (2014). An assembly sequence planning approach with binary gravitational search algorithm. New Trends in Software Methodologies, Tools and Techniques, IOS Press, pp. 179-193.

[34] Mukred, J.A.A., Ibrahim, Z., Ibrahim, I., Adam, A., Wan, K., Md Yusof, Z. and Mokhtar, N. (2012). A binary particle swarm optimization approach to optimize asembly sequence planning. Advanced Science Letters, vol. 13, pp. 732-738.

[35] Ibrahim, I., Ibrahim, Z., Ahmad, H. and Md Yusof, Z. (2016). An assembly sequence planning approach with a multi-state particle swarm optimization. Lecture Notes in Computer Science, vol. 9799, pp. 841-852.