mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4895

ORIGINAL ARTICLE

A Tutorial on Single-solution Simulated Kalman Filter

Nor Hidayati Abdul Aziz[1,2], Nor Azlina Ab. Aziz[2], Badaruddin Muhammad[1], Kamil Zakwan Mohd Azmi[1], Zulkifli Md Yusof[1], Mohd Saberi Mohamad[3], Mohd Ibrahim Shapiai[4] and Yusei Tsuboi[5]

[1]Universiti Malaysia Pahang, Pekan, Pahang, Malaysia.
[2]Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia.
[3]Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, Pengkalan Chepa, Kelantan, Malaysia.
[4]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.
[5]Vision Solution Center, SICK K.K., Tokyo, Japan.

ABSTRACT – Simulated Kalman Filter (SKF) is an estimation-based optimization algorithm which is established based on the Kalman filtering framework. A variant of SKF which operates using one agent is called single-solution simulated Kalman filter (ssSKF). At present, there is no tutorial been published on ssSKF. One may find that the equations and flowchart of the algorithm is not easy to understand. Hence, this paper provides a tutorial on ssSKF algorithm that emphasizes on a numerical example for easy and intuitive explanations. This tutorial would be important to those who work on the fundamentals and applications of ssSKF as well as to students who are new to optimization research.

ARTICLE HISTORY
Received: 2 April 2019
Accepted: 3 July 2019

KEYWORDS
Optimization
ssSKF
SKF

Introduction

The simulated Kalman filter (SKF) has been introduced in 2015 for numerical optimization problems [1-3]. It was introduced as population-based metaheuristics, where the search for optimal solution is conducted by a group of agents. The agents of SKF work like Kalman filters [4] and the measurement in SKF is a simulated measurement which is obtained using mathematical equation.

Many studies on SKF can be found in literature. For example, the SKF has been studied fundamentally [5-6]. The SKF also has been extended for binary optimization problems [7] and combinatorial optimization problems [8-10]. Hybridization of SKF with particle swarm optimization (PSO), gravitational search algorithm (GSA), and opposition-based learning [11-17] have also been proposed for better performance. Other variants called parameter-less SKF and randomized SKF algorithms were proposed in [18-19]. The SKF has also been applied for real world problems like the adaptive beamforming in wireless cellular communication [20-23], airport gate allocation problem [24-25], feature selection of EEG signal [26-27], system identification [28-29], image processing [30-31], controller tuning [32], and PCB drill path optimization [33].

A study in 2018 proved that the SKF algorithm able to operate using only one agent. This variant of SKF is called single-solution simulated Kalman filter (ssSKF) [34]. The ssSKF offers a slight advantage over the SKF counterpart in terms of the number of parameters in the algorithm. At present, the ssSKF has been applied in solving routing problem in printed circuit board drilling process [35].

This paper presents the first tutorial on ssSKF which emphasizes on the calculation aspect of ssSKF. This paper consist of two parts. The first part explains the fundamentals of the ssSKF while the second part shows a numerical example based on a function minimization problem.

The Single-solution Simulated Kalman Filter

The single-solution Simulated Kalman Filter (ssSKF) algorithm is a single agent version of the population-based Simulated Kalman Filter algorithm. Similar to SKF, the ssSKF algorithm attempts to solve optimization problems by iteratively estimating the optimum solution using the scalar model of discrete Kalman filter framework. By using this model, the state vector, $\boldsymbol{X}$, holds the agent's estimated position, which is a scalar value for each dimension in the search space. This estimated state variable is used in the calculation of fitness based on the specific objective function.

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

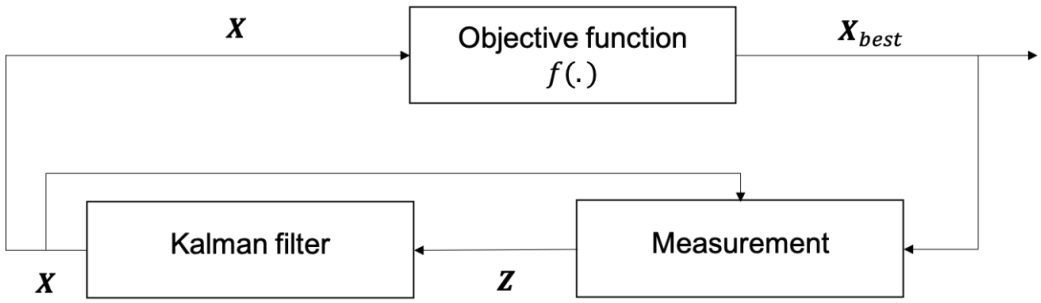

Figure 1. Principle of the ssSKF algorithm.

In ssSKF, the algorithm starts with the initialization of the single agent. This agent represents a Kalman Filter. Next, the fitness of the agent is evaluated. During each iteration, the best-so-far solution, $\boldsymbol{X}_{best}$, which holds the best-found solution so far, is updated. The single agent in ssSKF algorithm iteratively improves its estimation by using the standard Kalman Filter framework which comprises of predict, measure, and estimate phases. An adaptive neighbourhood is employed to make a prediction during the prediction phase. The measurement, guided by the best-so-far solution, $\boldsymbol{X}_{best}$, is simulated during the measurement phase. Finally, the agent makes an estimation of the optimum solution during the estimation phase, with the influence of the Kalman gain, K. This process continues until the stopping condition is met.

Figure 1 illustrates the basic principle of the ssSKF algorithm. This figure shows that principally, the ssSKF algorithm uses only a single agent which acts as a Kalman filter. The estimation by the agent is involved in the calculation of fitness by the objective function, after which the best-so-far solution, $\boldsymbol{X}_{best}$, is determined and updated. This best-so-far solution, $\boldsymbol{X}_{best}$, guides the simulated measurement process that provides a measurement to the Kalman filter to make an estimation for the next time step.

The flowchart of ssSKF algorithm is shown in Figure 2. The algorithm begins with random initial solution, $X(0)$. Initial error covariance, $P(0)$, is set to a normally distributed random number. Then, fitness is calculated. After that, according to the type of problem, the best-so-far solution, $\boldsymbol{X}_{best}$, is updated. The $\boldsymbol{X}_{best}$ is updated only if the solution at current iteration, $\boldsymbol{X(t)}$, is better. For minimization problem, $\boldsymbol{X}_{best}$ is updated if the fitness of $\boldsymbol{X(t)}$ is less than the fitness of $\boldsymbol{X}_{best}$. On the other hand, for maximization problem, $\boldsymbol{X}_{best}$ is updated if the fitness of $\boldsymbol{X(t)}$ is greater than the fitness of $\boldsymbol{X}_{best}$.

During prediction, the following equations are used to predict the optimum solution:

$$\boldsymbol{X}^d(t|t+1) \sim U\left[\boldsymbol{X}^d_{best} - \delta, \boldsymbol{X}^d_{best} + \delta\right] \quad (1)$$

$$P^d(t|t+1) = P^d(t) + randn^d \quad (2)$$

where $randn^d$ is a normally distributed random number and is used whenever the parameter value is needed for each dimension in every iteration. The ssSKF makes a prediction that resides in a confined neighborhood of $[\boldsymbol{X}^d_{best} - \delta,\ \boldsymbol{X}^d_{best} + \delta]$ in every dimension, with the $\boldsymbol{X}_{best}$ to be the center of the neighborhood. The size of the local neighbourhood is determined by the adaptively decreasing step-size δ.

$$\delta_t = e^{-\frac{\alpha \times t}{t_{Max}}} \times \delta_0 \quad (3)$$

where t_{Max} is the maximum number of iterations. The initial neighbourhood limit, δ_0, is determined using δ_0 = max(|lowerlimit|, |upperlimit|) to ensure maximum coverage of the search space during the first iteration.

Note that the ssSKF algorithm uses an adaptive neighbourhood in making prediction. The idea of having a prediction step is to make the best guess on the location of the optimal solution. This element is missing in the original SKF algorithm. In view of having $\boldsymbol{X}_{best}$ as the best-so-far solution, it is wise to predict that the position of the optimum solution is somewhere near $\boldsymbol{X}_{best}$. Therefore, in ssSKF, a decreasing local neighbourhood is adopted during the prediction step to further exploit this information.

To illustrate the concept of local neighbourhood in ssSKF algorithm, a two-dimensional problem of sphere function bounded by [-2,2] in both dimensions is used. The function is shown in (4).

$$\mathrm{f}(\boldsymbol{X}) = \sum_{j=1}^{2} x_j^2 = x_1^2 + x_2^2 \quad (4)$$

Figure 3 shows that the position of the optimum solution at $t = 1$ is predicted (marked in green) to be located in a confined neighborhood of $[X^d_{best} - \delta_t,\ X^d_{best} + \delta_t]$ which is marked by the black line, with the

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4280

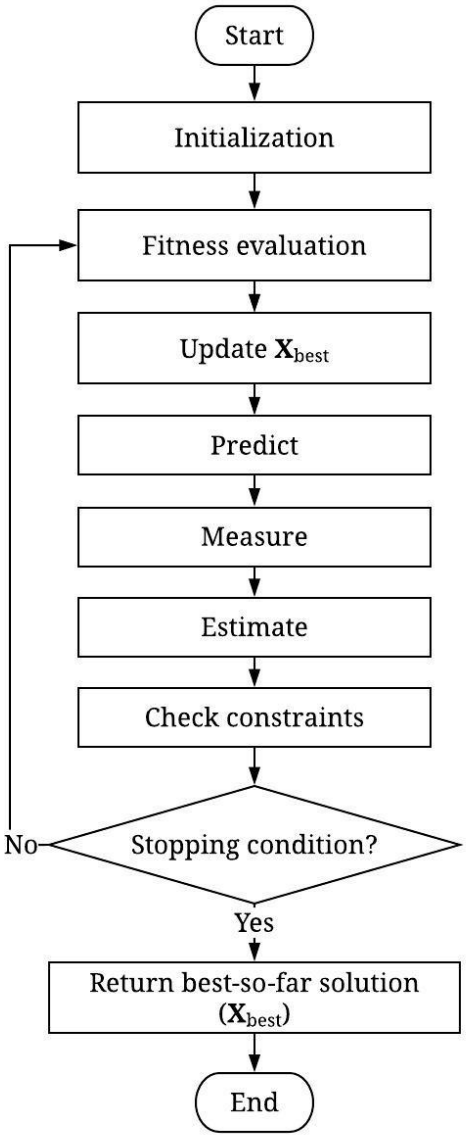

Figure 2. Flowchart of the ssSKF algorithm.

best-so-far solution, $\boldsymbol{X}_{best}$, at the center of the neighborhood. This local neighborhood, N_S, is decreasing in size, determined by the step-size, δ_t, as the iteration increases following (3). Figure 4 shows that the prediction of the optimum at iteration $t = 10$ happens in a smaller local neighbourhood. The size of local neighbourhood reflects that certainty in the prediction. As iteration increases, the fitness of the best-so-far solution is also improved. A smaller local neighbourhood during prediction indicates that it is almost certain that the optimum solution is located very near to the best-so-far solution, $\boldsymbol{X}_{best}$.

This certainty in prediction (reflected by the size of the local neighbourhood), however, cannot happen too early or it might lead to premature convergence. Figure 6 shows the plot of step-size, δ_t, for different values of the adaptive coefficient, α, over 100 iterations. It can be observed that a small value of α will lead to an almost linear decrement, while a larger value of α will lead to a faster convergence.

It is worth to note here that the decrement of the step size, δ_t, is also dependent on the maximum number of iterations. Since there is only one agent in ssSKF algorithm and there is only one function evaluation per agent per iteration, the step size, δ_t, can be said dependent on the maximum number of function evaluations. Simpler problems might not need a high number of function evaluations to reach ideal solution as compared to complex problems.

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4280

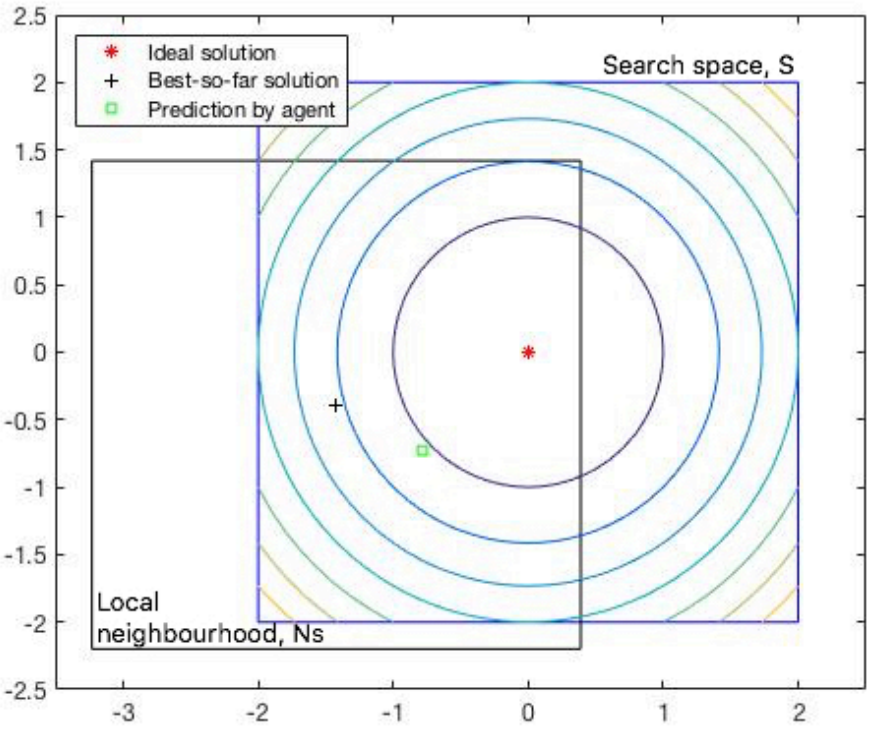

Figure 3. Local neighborhood during prediction ($t = 1$).

Also, the landscape of the problem also might influence the optimal value of the adaptive coefficient, α. For example, solving a unimodal problem might benefits from a high value of adaptive coefficient, but for solving a multimodal problem, a high value of adaptive coefficient might cause the optimization process to converge prematurely. The choice of this adaptive coefficient, α, is problem dependent. Thus, tuning is required to achieve the best solution.

The next step is measurement calculated at every dimension. In this step, the best-so-far solution, $\mathbf{X}_{best}^{d}$, steered the agent's simulated measurement value, $\mathbf{Z}^{d}(t)$, as follows:

$$\boldsymbol{Z}^{d}(t) = \boldsymbol{X}^{d}(t|t+1) + \Delta \quad (5)$$

where

$$\Delta = sin(rand^{d} \times 2\pi) \times \left|\boldsymbol{X}^{d}(t|t+1) - \boldsymbol{X}_{best}^{d}\right| \quad (6)$$

The purpose of the measurement is to give feedback to the estimation process. The measurement is simulated in such a way that the measured value of the agent may take any random value surrounding the predicted value, $\mathbf{X}^{d}(t|t+1)$, either approaching to or moving away from the best-so-far solution, $\mathbf{X}_{best}^{d}$, balancing between exploration and exploitation. The exploration and exploitation mechanisms are further compromised as the distance between the predicted value and the best-so-far solution decreases with the increase of the number of iterations.

Finally, during the estimation step, the solution and error covariance estimates for the next iteration are calculated using the estimate equations right after the calculation of the Kalman gain.

$$K^{d}(t) = \frac{P^{d}(t|t+1)}{P^{d}(t|t+1)+rand^{d}} \quad (7)$$

$$\mathbf{X}^{d}(t+1) = \mathbf{X}^{d}(t|t+1) + \gamma \quad (8)$$

$$\gamma = K^{d}(t) \times (\mathbf{Z}^{d}(t) - \mathbf{X}^{d}(t|t+1)) \quad (9)$$

$$P^{d}(t+1) = (1 - K^{d}(t)) \times P^{d}(t|t+1) \quad (10)$$

At the end of the estimation step, a better solution for the next iteration that lies between the predicted and the measured value may be produced. This process continues until the maximum number of iterations.

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

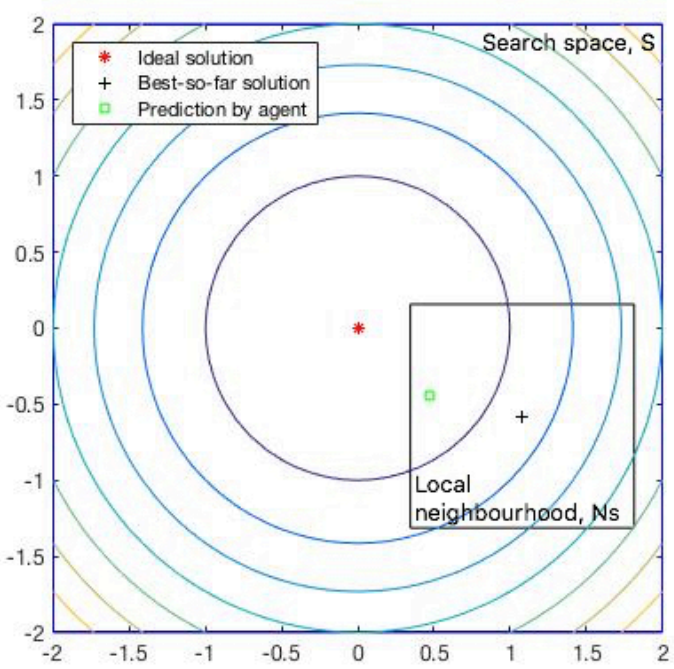

Figure 4. Local neighborhood (decreased in size) during prediction ($\boldsymbol{t = 10}$).

Numerical Example

In order to understand how the single-solution simulated Kalman filter (ssSKF) algorithm operates, consider a two-dimensional problem of sphere function bounded by [-2,2] in both dimensions. The function is similar to (4).

Figure 5 shows the three-dimensional view of the sphere function. The ideal solution for the given objective function is at the centre of the search space (0,0), where the fitness value is equal to 0 (minimization problem).

The agent of ssSKF is represented by a state vector of two dimensions, $\boldsymbol{X}(t) = \{x^1(t), x^2(t)\}$. For minimization problem, the fitness of the solution is first set to infinity, $fit(\boldsymbol{X}_{true}) = \infty$.

The first step is initialization. At $t = 0$, the initial estimated state of the agent, $\boldsymbol{X}(0)$, is distributed randomly in uniform distribution within the search space of [-2,2] in every dimension. A normally distributed random number, $randn_i^d$, defined in the range of (0,1) with a mean of 0.5, is specified in every dimension for the initial error covariance of each agent, $\boldsymbol{P}(0)$.

$$\boldsymbol{X(0)} = \{0.9271, -0.2500\}$$

$$\boldsymbol{P(0)} = \{0.5341, 0.5771\}$$

Figure 6 illustrates the position of the estimated state of the SKF agents during initialization at $t = 0$, on the contour plot of the sphere function's search space. The position of the ideal solution is marked by '*', while the position of agents is represented by square boxes.

In the second step, the fitness the agent is evaluated:

$$f(\boldsymbol{X}(0)) = 0.9271^2 + (-0.25)^2 = 0.9220$$

Then, based on the fitness values, the best-so-far solution, $\boldsymbol{X}_{best}$ is updated. In this specified iteration, it is found that the agent has better fitness value ($0.9220 < \infty$), thus, the best value is updated.

$$\boldsymbol{X}_{best} = \{0.9271, -0.2500\}$$

Figure 7 shows the $\boldsymbol{X}_{best}$ update after the fitness evaluation step.

The third step start with prediction phase. This is calculated using (3). In ssSKF, optimum solution is predicted to be located in the local neighbourhood, N_s, surrounding the best-so-far solution, X_{best}^d.

To do the prediction, first, we need to calculate the step size, δ. Let the adaptive coefficient, α equals to 10, and the maximum iteration to be 10. The initial step size depends on the size of the search space, $\delta_0 = \max(|-2|, |2|) = 2$. Thus, the step size, δ, and the corresponding predicted state estimate for the first iteration is:

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4280

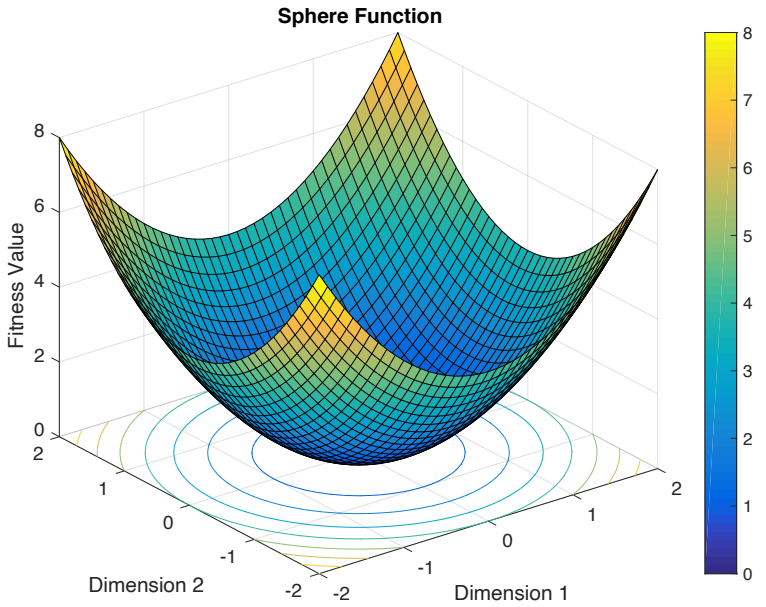

Figure 5. Three-dimensional view of sphere function.

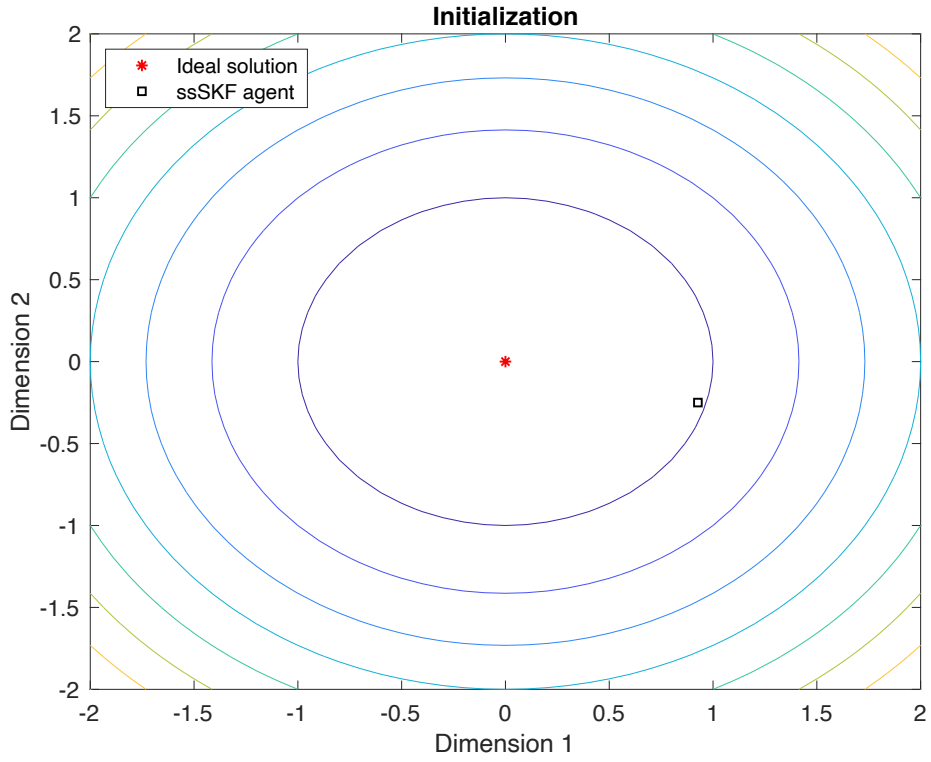

Figure 6. Estimated position by the ssSKF agent in the search space (initialization).

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4280

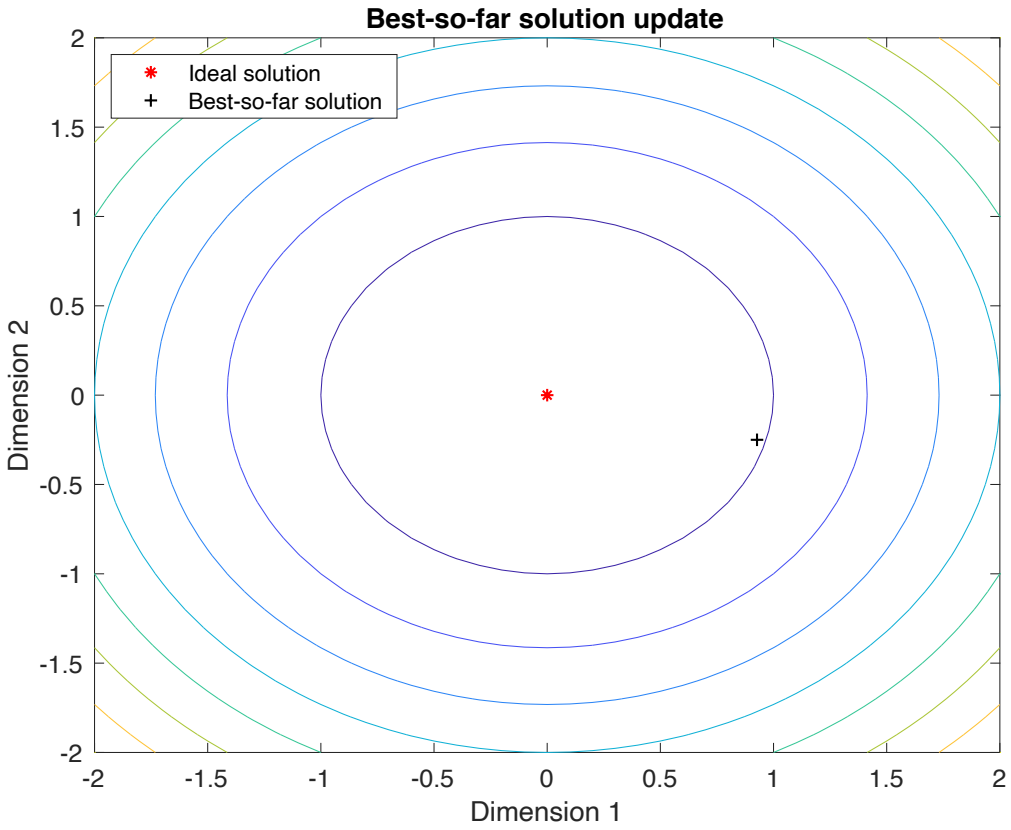

Figure 7. Best-so-far solution ($\boldsymbol{X_{best}}$) update.

$$\delta_1 = \mathrm{e}^{-\frac{10\times1}{100}} \times 2 = 1.8097$$

$$X^d(0|1) \sim U\left[X^d_{best} - \delta_1, X^d_{best} + \delta_1\right] = \{-0.0639, -0.6676\}$$

Figure 8 shows the predicted position of the optimal solution by the ssSKF agent is located inside the local neighbourhood.

The error covariance is predicted to be influenced by the process noise. A normally distributed random number, $randn_i^d$, defined in the range of (0,1) with a mean of 0.5, is specified in every dimension as the process noise of each agent, $\boldsymbol{Q}(0)$. Let the process noise for each agent, $\boldsymbol{Q}(0)$, be:

$$\boldsymbol{Q}(0) = \{0.4467, 0.5542\}$$

$$\boldsymbol{P}(0|1) = \boldsymbol{P}(0) + \boldsymbol{Q}(0) = \{0.5341 + 0.4467, 0.5771 + 0.5542\} = \{0.9808, 1.1313\}$$

The prediction phase is followed by the simulated measurement phase. The random number, $rand^d$, used in measurement based on (5) are taken from a uniform distribution in the range of (0,1). Let the random number, $rand$ be:

$$\boldsymbol{rand} = \{0.2240, 0.1014\}$$

$$Z^1(0) = X^1(0|1) + sin(rand^1 \times 2\pi) \times \left|X^1(0|1) - X^1_{best}\right|$$

$$= -0.0639 + sin(0.2240 \times 2\pi) \times |-0.0639 - 0.9271|$$

$$= 0.9139$$

$$Z^2(0) = X^2(0|1) + sin(rand^2 \times 2\pi) \times \left|X^2(0|1) - X^2_{best}\right|$$

$$= -0.6676 + sin(0.1014 \times 2\pi) \times |-0.6676 - (-0.2500)|$$

$$= -0.4192$$

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

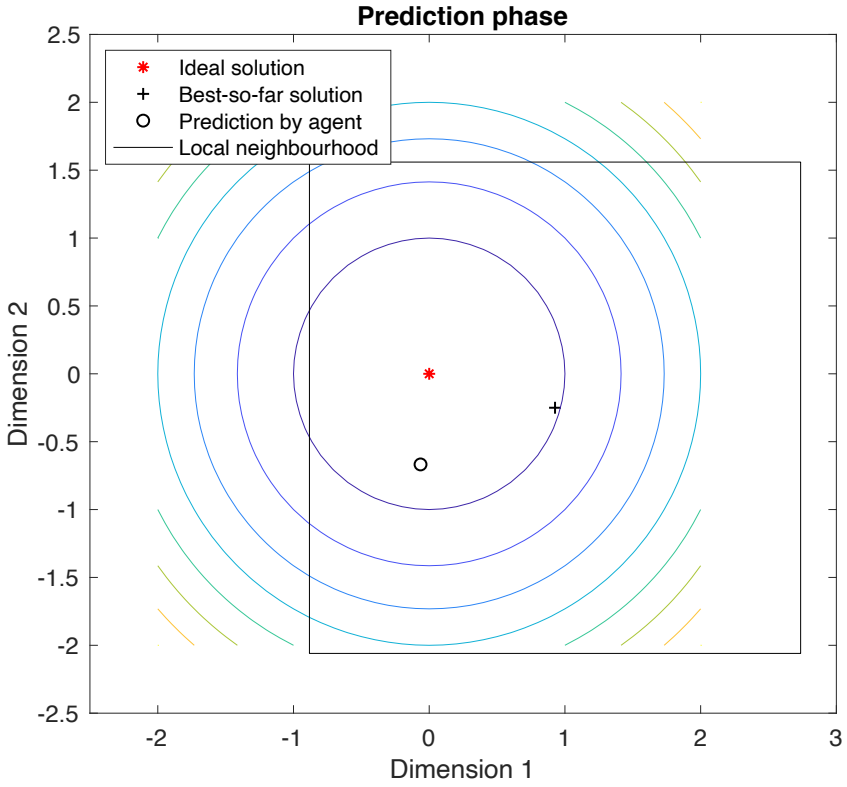

Figure 8. Predicted position by the ssSKF agent in the search space during prediction phase.

Figure 9 shows the simulated measurement value for each agent and their corresponding range. The effect of the sine function is to provide a balance between exploration and exploitation during the simulated measurement process while allowing more possibility at the extreme values. A simulated measurement may take any value bounded by the distance between the predicted state estimate to the best-so-far solution, $\boldsymbol{X}_{best}$ in both dimensions. The farther predicted value from $\boldsymbol{X}_{best}$, the bigger the range. This allows more exploration of the search space by the agent.

Lastly, estimation for the next time step is carried out by calculations based on (7) to (10). The estimation phase is preceded by calculation of Kalman gain. A normally distributed random number, $randn^d$, defined in the range of (0,1) with a mean of 0.5, is specified in every dimension as the measurement noise of the agent, $\boldsymbol{R}(0)$. Let the measurement noise for the agent, $\boldsymbol{R}(0)$, be:

$$\boldsymbol{R(0)} = \{0.6242, 0.4868\}$$

$$K^1(0) = \frac{P^1(0|1)}{\left(P^1(0|1) + R^1(0)\right)} = \frac{0.9808}{0.9808 + 0.6242}$$

$$= 0.6111$$

$$X^1(1) = X^1(0|1) + K^1(0) \times \left(Z^1(0) - X^1(0|1)\right)$$

$$= -0.0639 + 0.6111 \times \left(0.9139 - (-0.0639)\right)$$

$$= 0.5336$$

$$P_1^1(1) = \left(1 - K^1(0)\right) \times P^1(0|1)$$

$$= (1 - 0.6111) \times 0.9808$$

$$= 0.3814$$

$$K^2(0) = \frac{P^2(0|1)}{\left(P^2(0|1) + R^2(0)\right)}$$

$$= \frac{1.1313}{1.1313 + 0.4868}$$

$$= 0.6992$$

$$X^2(1) = X^2(0|1) + K^2(0) \times \left(Z^2(0) - X^2(0|1)\right)$$

$$= -0.6676 + 0.6992 \times \left(-0.4192 - (-0.6676)\right)$$

$$= -0.4939$$

$$P^2(1) = \left(1 - K^2(0)\right) \times P^2(0|1)$$

$$= (1 - 0.6992) \times 0.6992 = 0.3403$$

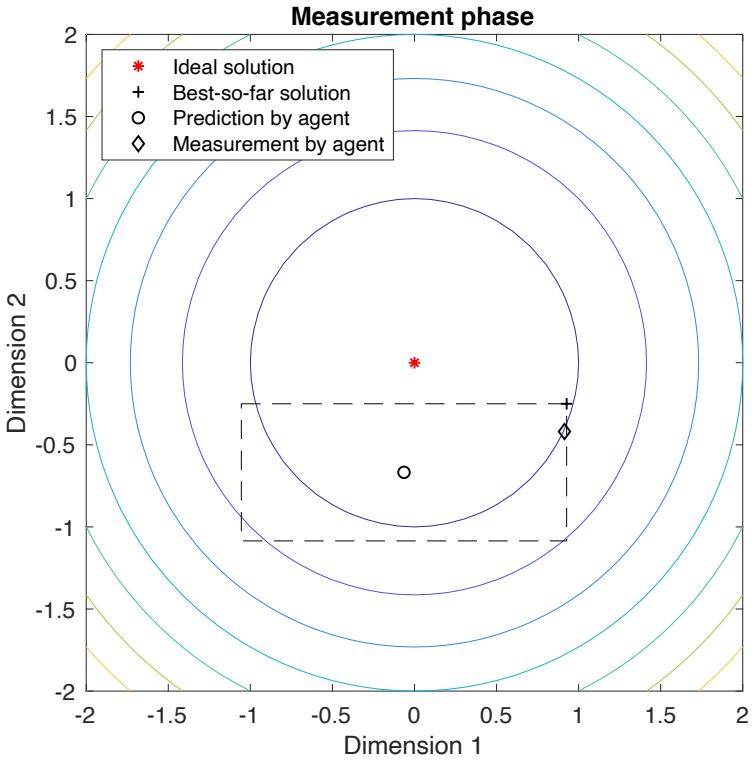

Figure 9. Simulated measurement value by the ssSKF agent during the measurement phase.

Figure 10 shows the estimation position of the optimum solution by the ssSKF agent during the estimation phase. Figure 11 on the other hand, shows the estimated position of the optimal solution at $t = 0$ and at $t = 15$. It can be seen that the estimation by the ssSKF agent has improved during the search. Finally, these steps will be repeated until the stopping condition is met.

Table 1 gives a summary of the agent' predict, measure and estimate values from $t = 1$ to $t = 15$ with their corresponding estimation fitness value and best-so-far solution.

Conclusions

The ssSKF algorithm is based on Kalman filtering computation in finding the global minimum/maximum for numerical optimization problems. This set of computation is different compared to other well-established optimization algorithms. Students and those who have no experience working on Kalman filtering might have difficulties in the implementation of SKF in MATLAB or other comparable software to solve an optimization problem. This tutorial is fruitful for them to understand the calculation involved in ssSKF.

Acknowledgement

This research is supported by the Fundamental Research Grant Scheme 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] 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, vol. 9, pp. 3415-3422.

[4] Kalman, R.E. (1960) A new approach to linear filtering and prediction problems. ASME Journal of Basic Engineering, vol. 82, pp. 35-45.

[5] 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.

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4280

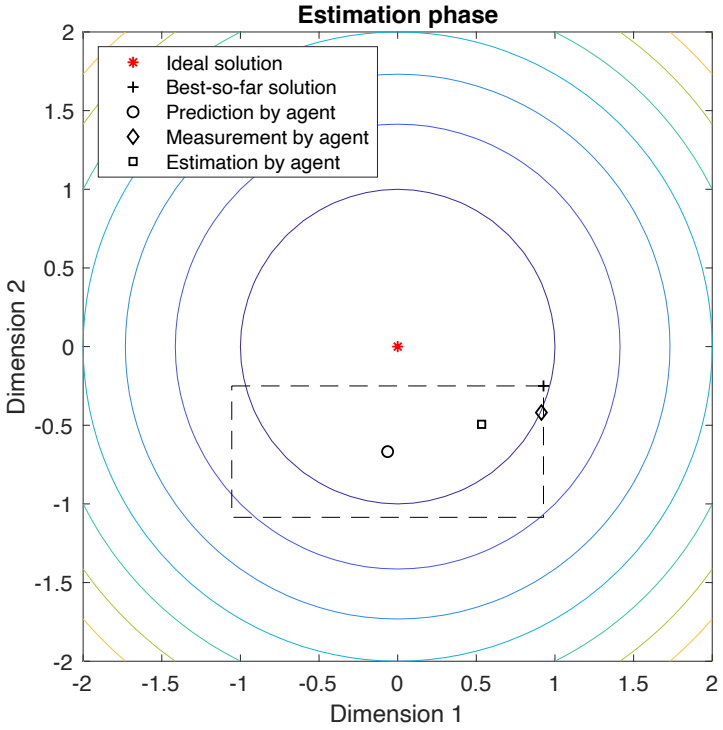

Figure 10. Estimated position by the ssSKF agent during the estimation phase.

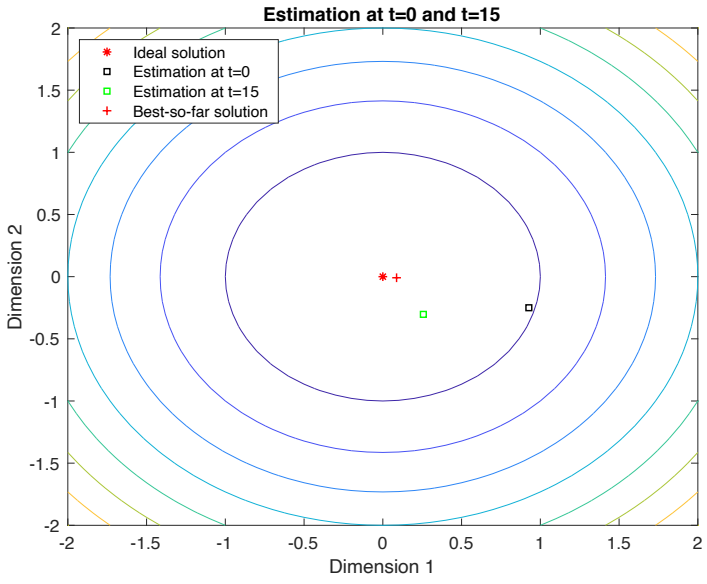

Figure 11. Estimated position of the optimum solution in the search space at $t = 0$ and $t = 15$.

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 33 – 44
DOI: https://doi.org/10.15282/mekatronika.v1i2.4280

Table 1. Summary of ssSKF predict, measure and estimate values from iteration 1 to 15.

Iter. No.	Predict	Measure	Estimate	Reinitialize	Fitness	X_{best}
Iter. 1	{-0.0639, -0.6676}	{0.9139, -0.4192}	{0.5336, -0.4939}	-	0.5287	{0.9271, -0.2500}
Iter. 2	{2.1143, 0.2601}	{3.4279, -0.1199}	{2.9577, 0.0109}	{-0.7491, 0.0109}	0.5613	{0.5336, -0.4939}
Iter. 3	{-0.4695, -1.4458}	{-0.0022 -0.9146}	{-0.1927, -1.1289}	-	1.3116	{0.5336, -0.4939}
Iter. 4	{0.9051, -1.7445}	{1.0607, -0.6116}	{1.0062, -1.0720}	-	2.1616	{0.5336, -0.4939}
Iter. 5	{1.0634, 0.6431}	{0.6431, 1.6557}	{1.0108, 1.3132}	-	2.7463	{0.5336, -0.4939}
Iter. 6	{-0.3646, -1.0066}	{0.3740, -0.5036}	{0.0889, -0.7121}	-	0.5149	{0.5336, -0.4939}
Iter. 7	{0.1327, 0.3642}	{0.0993, 0.0815}	{0.1125, 0.1915}	-	0.0493	{0.0889, 0.7121}
Iter. 8	{0.4633, -0.5850}	{0.8138, 0.1811}	{0.6891, -0.1006}	-	0.4849	{0.1125, 0.1915}
Iter. 9	{0.3977, -0.6108}	{0.2270, -0.0870}	{0.2981, -0.2642}	-	0.1587	{0.1125, 0.1915}
Iter. 10	{0.5315, 0.1493}	{0.6249, 0.1870}	{0.5998, 0.1738}	-	0.3900	{0.1125, 0.1915}
Iter. 11	{0.2555, -0.2190}	{0.1129, 0.1910}	{0.1624, 0.0568}	-	0.0296	{0.1125, 0.1915}
Iter. 12	{-0.0936, -0.4359}	{-0.2116, -0.8859}	{-0.1637, -0.6727}	-	0.4793	{0.1624, 0.0568}
Iter. 13	{0.3575, 0.2048}	{0.2916, 0.3479}	{0.3120, 0.2884}	-	0.1806	{0.1624, 0.0568}
Iter. 14	{0.1132, 0.0160}	{0.0714, -0.0243}	{0.0872, -0.0088}	-	0.0077	{0.1624, 0.0568}
Iter. 15	{0.3844, -0.2262}	{0.1900, -0.3336}	{0.2573, -0.3032}	-	0.1581	{0.0872, 0.0088}

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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

*CORRESPONDING AUTHOR | Nor Hidayati Abdul Aziz | ✉ hidayati.aziz@mmu.edu.my

filter (SKF) with gravitational search algorithm (GSA). The National Conference for Postgraduate Research, pp. 854-864.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] Muhammad, B., Mohd Azmi, K.Z., Ibrahim, Z., Mohd Faudzi, A.A., and Pebrianti, D. (2018). Simultaneous computation of model order and parameter estimation for system identification based on opposition-based simulated Kalman filter. SICE International Symposium on Control Systems 2018, pp. 105-112.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] Abdul 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.

[34] Abdul Aziz, N.H., Ibrahim, Z., Ab. Aziz, N.A, Mohamad, M.S., and Watada, J. (2018). Single-solution simulated Kalman filter algorithm for global optimisation problems. Sadhana, vol. 43, issue 7, article 103.

[35] 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.