mekatronika – Journal of Intelligent Manufacturing & Mechatronics
VOL. 01, ISSUE 02, 23 – 32
DOI: https://doi.org/10.15282/mekatronika.v1i2.4894

ORIGINAL ARTICLE

A Tutorial on Population-based Simulated Kalman Filter

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

[1]Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, Pekan, Pahang, Malaysia.
[2]Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia.
[3]Faculty of Electrical and Electronic Engineering, Universiti Malaysia Pahang, Pekan, Pahang, Malaysia.
[4]Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, Pengkalan Chepa, Kelantan, Malaysia.
[5]Vision Solution Center, SICK KK, Tokyo, Japan.

ABSTRACT – Simulated Kalman Filter (SKF) is an estimation-based optimization algorithm which is established based on the Kalman filtering framework. Even since the SKF algorithm is introduced in 2015, there is no tutorial been published on SKF. One may find that the equations and flowchart of the algorithm is not easy to understand. Hence, this paper provides a tutorial on SKF 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 SKF as well as to students who are new to optimization research.

ARTICLE HISTORY
Received: 7 April 2019
Accepted: 1 Jun 2019

KEYWORDS
Optimization
Simulated Kalman Filter

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], where they go through prediction, measurement, and estimation process in every iteration. 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 [33-34].

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

The Simulated Kalman Filter

The simulated Kalman filter (SKF) algorithm starts with the initialization of the population. Then, the solutions of the initial population are evaluated, and the best-so-far solution is updated. Next, the SKF algorithm iteratively improves its estimation by using the standard Kalman filter framework which comprises of predict, measure, and estimate. For a bounded constraint problem, if the estimated value falls outside the search space, its value will be re-initialized. This process continues until the stopping condition is met. The flowchart of the SKF algorithm is illustrated in Figure 1.

The SKF algorithm starts with the random initialization of its agents' estimated state within the search space, to produce the initial solution, $\boldsymbol{X}(0)$. The initial estimated state of each agent, $\boldsymbol{X}_i(0)$, is distributed randomly in uniform distribution within the search space 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, d, for the initial error covariance of each agent, $P_i(0)$. Last but not least, the maximum number of iterations, $tMax$, is initialized based on the maximum number of function evaluations.

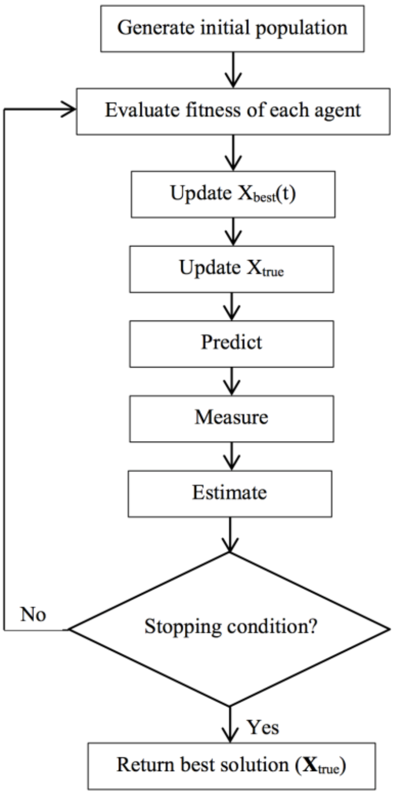

Figure 1. The flowchart of the SKF.

The iteration begins with fitness evaluation of the N agents, $fit(X(t))$, where $X(t) = \{X_1(t), X_2(t), \ldots, X_i(t), \ldots, X_N(t)\}$, where t is the iteration number. Then, the best solution in the corresponding iteration, $X_{best}(t)$, is updated according to the type of the optimization problem. In minimization problem, $X_{best}(t)$ will assume the position of the agent with the minimum fitness in the corresponding iteration, whereas, for maximization problem, $X_{best}(t)$ will assume the position of the agent with the maximum fitness in the corresponding iteration.

After that, the best-so-far solution called as the true value, X_{true}, is updated. The true value, X_{true}, is updated only if a better solution is found, mathematically $fit(X_{best}(t)) < fit(X_{true})$ for minimization problem or $fit(X_{best}(t)) < fit(X_{true})$ for the maximization problem.

Then, the best solution in the corresponding iteration, $X_{best}(t)$, is updated according to the type of the optimization problem. In minimization problem, $X_{best}(t)$ will assume the position of the agent with the minimum fitness in the corresponding iteration, whereas, for maximization problem, $X_{best}(t)$ will assume the position of the agent with the maximum fitness in the corresponding iteration.

The search strategy in SKF algorithm has three phases; predict-measure-estimate. During the prediction phase, the current predicted state, $X_i(t|t+1)$, is assumed to be the estimated value:

$$X_i(t|t+1) = X_i(t) \tag{1}$$

This equation implies that the optimum solution is predicted to be located at the previously estimated position. The predicted error covariance for each agent, $P(t|t+1)$, on the other hand, is predicted to be influenced by the process noise, Q. Hence, the error covariant is predicted as follows:

$$P(t|t+1) = P(t) + Q \tag{2}$$

where Q is called process noise. In this paper, $Q = rand$. Meaning that a normally distributed random number, $rand$, defined in the range of (0,1) with a mean of 0.5, is specified in every dimension as the process noise and it is added to the currrent error covariance, $P(t)$.

In SKF, measurements are simulated using an agent's prediction and X_{true}. The dimensional wise calculation of measured value for each dimension of agent ith is calculated as follows:

$$Z_i(t) = X_i(t|t+1) + \sin(2\pi rand_i(t)) \times |X_i(t|t+1) - X_{true}| \tag{3}$$

where $rand_i(t)$ is a random value within the range of [0,1]. The estimation phase follows the measurement phase and the estimated next value is updated using (4):

$$X_i(t+1) = X_i(t|t+1) + K(t) \times (Z_i(t) - X_i(t|t+1)) \tag{4}$$

where $K(t)$ is the Kalman gain, which is calculated as follows:

$$K(t) = P(t|t+1)/(P(t|t+1) + R) \tag{5}$$

where $R=randn$ is the measurement noise. Then, the current error covariant estimate is updated in estimation phase using (6):

$$P(t+1) = (1-K(t)) \times P(t|t+1) \tag{6}$$

These steps continue until at the end of the iteration or at the end of the fitness evaluation.

Numerical Example

Consider a simple two-dimensional sphere function given by (7).

$$f(\mathbf{X}) = x_1^{\ 2} + x_2^{\ 2} \tag{7}$$

For simplicity, consider the test function is bounded in both dimensions by [-2,2]. Figure 2 shows the three-dimensional view of the this function. The ideal solution for the given objective function is at the center of the search space (0,0), where the fitness value is equal to 0 (minimization problem).

In this example, to illustrate how the SKF algorithm operates, three agents are used. Each agent i is represented by a state vector of two dimensions, $\boldsymbol{X}_i\ (t) = \{x_i^1 + x_i^2\}$. For minimization problem, the fitness of the solution is first set to infinity, fit($\boldsymbol{X}_{\text{true}}$) = ∞.

The first step is initialization. At t = 0, the initial estimated state of each agent, $\boldsymbol{X}_i(0)$, is distributed randomly in uniform distribution within the search space of [-2,2] in every dimension. A normally distributed random number, *randn*, 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, $\boldsymbol{P}_i(0)$.

$$\boldsymbol{X}_1(0) = \{0.9271, -0.2500\}$$

$$\boldsymbol{P}_1(0) = \{0.5341, 0.5771\}$$

$$\boldsymbol{X}_2(0) = \{1.7209, 0.5351\}$$

$$\boldsymbol{P}_2(0) = \{0.2597, 0.3414\}$$

$$\boldsymbol{X}_3(0) = \{-0.7373, -0.6772\}$$

$$\boldsymbol{P}_3(0) = \{0.3043, 0.3531\}$$

Figure 3 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 of each agent is first evaluated using (7).

$$f(\boldsymbol{X}_1(0)) = 0.9271^2 + (-0.25)^2 = 0.9220$$

$$f(\boldsymbol{X}_2(0)) = 1.7209^2 + 0.5351^2 = 3.2478$$

$$f(\boldsymbol{X}_3(0)) = (-0.7373)^2 + (-0.6772)^2 = 1.0022$$

Then, based on the fitness values, $\boldsymbol{X}_{\text{best}}(0)$ and $\boldsymbol{X}_{\text{true}}$ are determined and updated. In this specified iteration, it is found that the first agent has the most minimum fitness in the corresponding iteration, thus it is designated as $\boldsymbol{X}_{\text{best}}(0)$. And since its fitness is the best fitness found so far (0.9220 < ∞), thus, the true value is updated ($\boldsymbol{X}_{\text{true}} = \boldsymbol{X}_{\text{best}}(0)$). Figure 4 shows the $\boldsymbol{X}_{\text{true}}$ after fitness evaluation step.

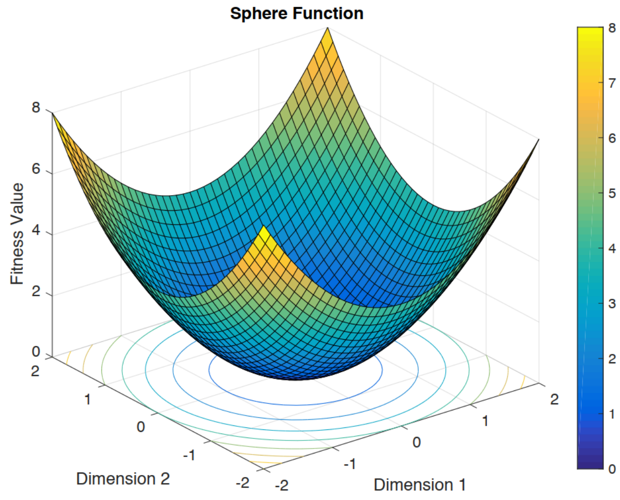

Figure 2. Three-dimensional view of sphere function.

The third step start with prediction phase. In SKF, the state prediction follows the last estimated state, while the error covariance is predicted to be influenced by the process noise. A normally distributed random number, *randn*, 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, $\boldsymbol{Q}_i(0)$. Let the process noise for each agent, $\boldsymbol{Q}_i(0)$, be:

$$\boldsymbol{Q}_1(0) = \{0.4467, 0.5542\}$$

$$\boldsymbol{Q}_2(0) = \{0.5448, 0.5806\}$$

$$\boldsymbol{Q}_3(0) = \{0.3105, 0.6273\}$$

The predicted state estimate and the predicted error covariance for each agent are calculated as follows:

$$\boldsymbol{X}_1(0|1) = \boldsymbol{X}_1(0) = \{0.9271, -0.2500\}$$

$$\boldsymbol{P}_1(0|1) = \boldsymbol{P}_1(0) + \boldsymbol{Q}_1(0) = \{0.5341 + 0.4467, 0.5771 + 0.5542\}$$
$$= \{0.9808, 1.1313\}$$

$$\boldsymbol{X}_2(0|1) = \boldsymbol{X}_2(0) = \{1.7209, 0.5351\}$$

$$\boldsymbol{P}_2(0|1) = \boldsymbol{P}_2(0) + \boldsymbol{Q}_2(0) = \{0.2597 + 0.5448, 0.3414 + 0.5806\}$$
$$= \{0.8045, 0.9220\}$$

$$\boldsymbol{X}_3(0|1) = \boldsymbol{X}_3(0) = \{-0.7373, -0.6772\}$$

$$\boldsymbol{P}_3(0|1) = \boldsymbol{P}_3(0) + \boldsymbol{Q}_3(0) = \{0.3043 + 0.3105, 0.3531 + 0.6273\}$$
$$= \{0.6148, 0.9804\}$$

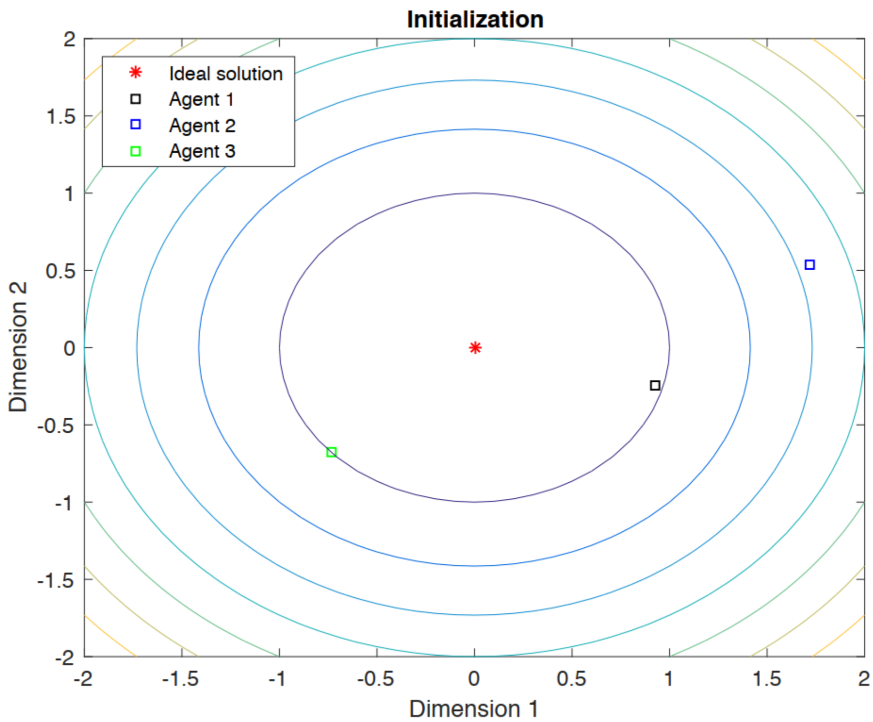

Figure 3. Estimated position by SKF agents in the search space (initialization).

$$\boldsymbol{X}_{best}(0) = \{0.9271, -0.2500\}, \qquad \boldsymbol{X}_{true} = \{0.9271, -0.2500\}$$

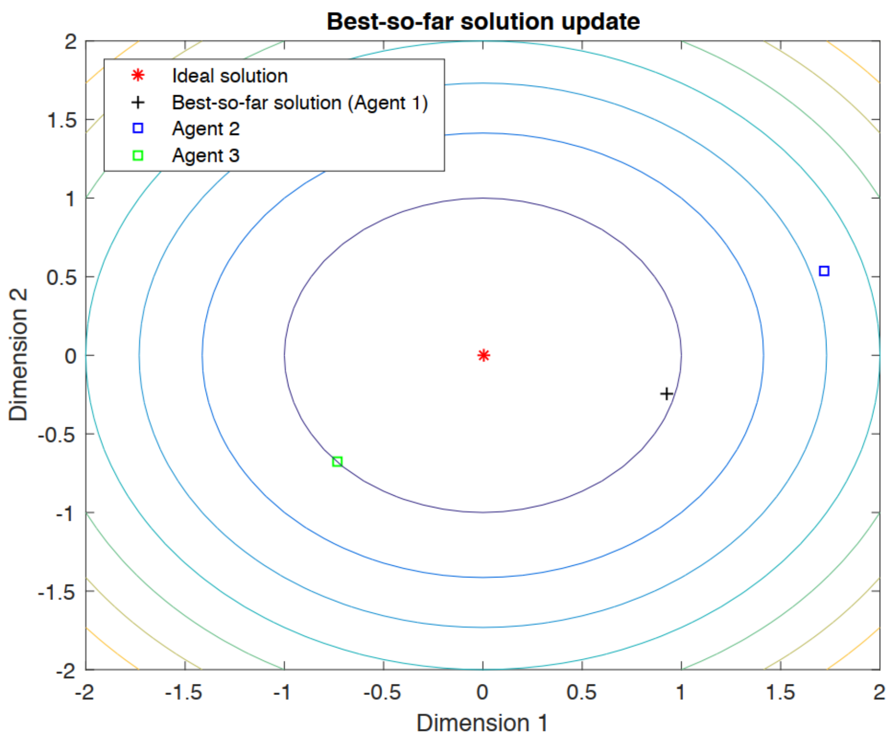

Figure 4. Best-so-far solution ($\boldsymbol{X}_{\text{true}}$) update.

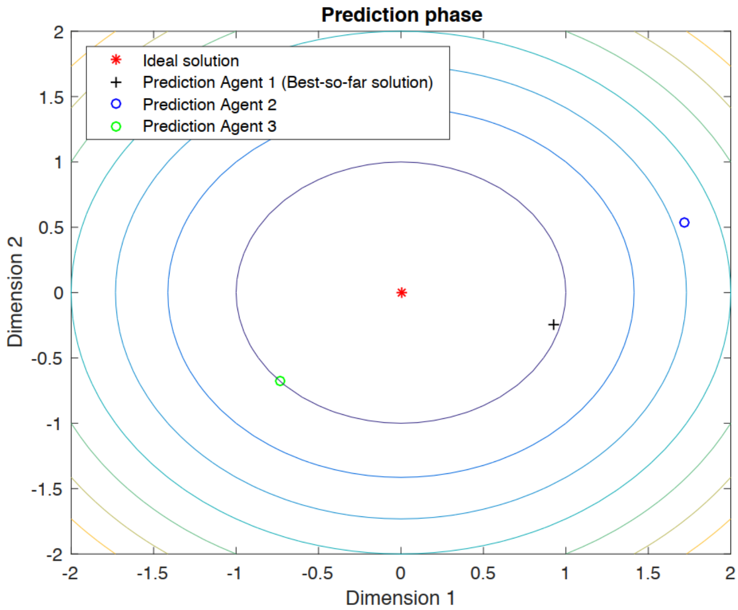

Figure 5. Predicted position by SKF agents in the search space during prediction phase.

Figure 5 shows the predicted position of the optimal solution by the SKF agents is the position of the previously estimated state, which is, in this case, is the initial state estimate. The prediction phase is followed by the simulated measurement phase. Equation (3) is used to get the simulated measurement value for each agent. Let the random number for each agent, ***rand***$_i$, be:

$$\boldsymbol{rand}_1 = \{0.2240, 0.1014\}$$

$$\boldsymbol{rand}_2 = \{0.2256, 0.1821\}$$

$$\boldsymbol{rand}_3 = \{0.7968, 0.2621\}$$

Hence,

$$Z_1^1(0) = X_1^1(0|1) + \sin(rand_1^1 \times 2\pi) \times |X_1^1(0|1) - X_{true}|$$
$$= 0.9271 + \sin(0.2240 \times 2\pi) \times |0.9271 - 0.9271| = 0.9271$$

$$Z_1^2(0) = X_1^2(0|1) + \sin(rand_1^2 \times 2\pi) \times |X_1^2(0|1) - X_{true}|$$
$$= -0.2500 + \sin(0.1014 \times 2\pi) \times |-0.2500 - (-0.2500)|$$
$$= -0.2500$$

$$Z_2^1(0) = X_2^1(0|1) + \sin(rand_2^1 \times 2\pi) \times |X_2^1(0|1) - X_{true}| =$$
$$= 1.7209 + \sin(0.2256 \times 2\pi) \times |1.7209 - 0.9271| = 2.5054$$

$$Z_2^2(0) = X_2^2(0|1) + \sin(rand_2^2 \times 2\pi) \times |X_2^2(0|1) - X_{true}|$$
$$= 0.5351 + \sin(0.1821 \times 2\pi) \times |\, 0.5351 - (-0.2500)| = 1.2498$$

$$Z_3^1(0) = X_3^1(0|1) + \sin(rand_3^1 \times 2\pi) \times |X_3^1(0|1) - X_{true}| =$$
$$= -0.7373 + \sin(0.7968 \times 2\pi) \times |-0.7373 - 0.9271| = -2.3303$$

$$Z_3^2(0) = X_3^2(0|1) + \sin(rand_3^2 \times 2\pi) \times |X_3^2(0|1) - X_{true}|$$
$$= -0.6772 + \sin(0.2621 \times 2\pi) \times |-0.6772 - (-0.2500)|$$
$$= -0.2512$$

Figure 6 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, $\boldsymbol{X}_{\text{true}}$, in both dimensions. The farther predicted value from the $\boldsymbol{X}_{\text{true}}$, the bigger the range. This allows more exploration of the search space by the agents. The simulated measurement value for the first agent is its own position because it holds the best fitness so far.

Lastly, estimation for the next time step is carried out by calculations based on (4). The estimation phase is preceded by calculation of Kalman gain using (5). A normally distributed random number, *randn*, defined in the range of (0,1) with a mean of 0.5, is specified in every dimension as the measurement noise of each agent, $\boldsymbol{R}_i(0)$. Let the measurement noise for each agent, $\boldsymbol{R}_i(0)$, be:

$$\boldsymbol{R}_1(0) = \{0.6242, 0.4868\}$$

$$\boldsymbol{R}_2(0) = \{0.2677, 0.3671\}$$

$$\boldsymbol{R}_3(0) = \{0.3548, 0.3548\}$$

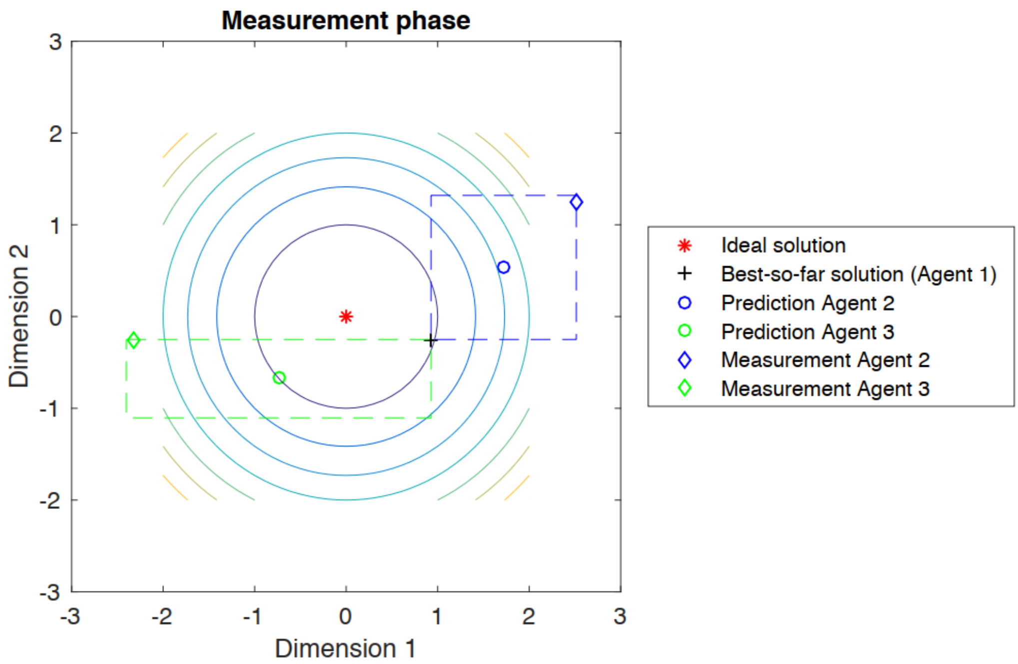

Figure 6. Simulated measurement value of SKF agents during the measurement phase.

Thus,

$$K_1^1(0) = \frac{P_1^1(0|1)}{(P_1^1(0|1) + R_1^1(0))} = \frac{0.9808}{0.9808 + 0.6242} = 0.6111$$

$$X_1^1(1) = X_1^1(0|1) + K_1^1(0) \times \left(Z_1^1(0) - X_1^1(0|1)\right)$$
$$= 0.9271 + 0.6111 \times (0.9271 - 0.9271) = 0.9271$$

$$P_1^1(1) = \left(1 - K_1^1(0)\right) \times P_1^1(0|1) = (1 - 0.6111) \times 0.9808 = 0.3814$$

$$K_1^2(0) = \frac{P_1^2(0|1)}{(P_1^2(0|1) + R_1^2(0))} = \frac{1.1313}{1.1313 + 0.4868} = 0.6992$$

$$X_1^2(1) = X_1^2(0|1) + K_1^2(0) \times \left(Z_1^2(0) - X_1^2(0|1)\right)$$
$$= -0.2500 + 0.6992 \times \left(-0.2500 - (-0.2500)\right) = -0.2500$$

$$P_1^2(1) = \left(1 - K_1^2(0)\right) \times P_1^2(0|1) = (1 - 0.6992) \times 0.6992 = 0.3403$$

$$K_2^1(0) = \frac{P_2^1(0|1)}{(P_2^1(0|1) + R_2^1(0))} = \frac{0.8045}{0.8045 + 0.2677} = 0.7503$$

$$X_2^1(1) = X_2^1(0|1) + K_2^1(0) \times \left(Z_2^1(0) - X_2^1(0|1)\right)$$
$$= 1.7209 + 0.7503 \times (2.5054 - 1.7209) = 2.3095$$

$$P_2^1(1) = \left(1 - K_2^1(0)\right) \times P_2^1(0|1) = (1 - 0.6111) \times 0.9808 = 0.2009$$

$$K_2^2(0) = \frac{P_2^2(0|1)}{(P_2^2(0|1) + R_2^2(0))} = \frac{0.9220}{0.9220 + 0.3671} = 0.7152$$

$$X_2^2(1) = X_2^2(0|1) + K_2^2(0) \times \left(Z_2^2(0) - X_2^2(0|1)\right)$$
$$= 0.5351 + 0.7152 \times (1.2498 - 0.5351) = 1.0463$$

$$P_2^2(1) = \left(1 - K_2^2(0)\right) \times P_2^2(0|1) = (1 - 0.7152) \times 0.9220 = 0.2626$$

$$K_3^1(0) = \frac{P_3^1(0|1)}{(P_3^1(0|1) + R_3^1(0))} = \frac{0.6148}{0.6148 + 0.3548} = 0.6341$$

$$X_3^1(1) = X_3^1(0|1) + K_3^1(0) \times \left(Z_3^1(0) - X_3^1(0|1)\right)$$
$$= -0.7373 + 0.6341 \times \left(-2.3303 - (-0.7373)\right) = -1.7474$$

$$P_3^1(1) = \left(1 - K_3^1(0)\right) \times P_3^1(0|1) = (1 - 0.641) \times 0.6148 = 0.2250$$

$$K_3^2(0) = \frac{P_3^2(0|1)}{(P_3^2(0|1) + R_3^2(0))} = \frac{0.9804}{0.9804 + 0.3816} = 0.7198$$

$$X_3^2(1) = X_3^2(0|1) + K_3^2(0) \times \left(Z_3^2(0) - X_3^2(0|1)\right)$$
$$= -0.6772 + 0.7198 \times \left(-0.2512 - (-0.6772)\right) = -0.3706$$

$$P_3^2(1) = \left(1 - K_3^2(0)\right) \times P_3^2(0|1) = (1 - 0.7198) \times 0.9804 = 0.2747$$

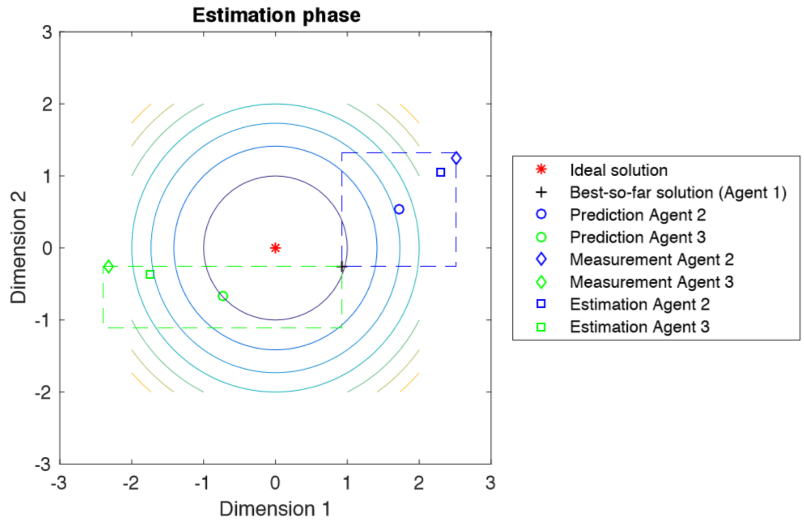

Figure 7. Estimated position by SKF agents during the estimation phase.

Figure 7 shows the estimation position of the optimum solution by the SKF agents during the estimation phase. Since the location of the second agent is outside the search space, $\boldsymbol{X}_2(1)$ = {2.3095,1.0463}, the value of the first dimension of the second agent, $X_2^1(1)$, is reinitialized randomly to be within the search space. Thus, the estimation for the next time step (after satisfying the boundary constraints) for all the three agents are:

$$\boldsymbol{X}_1(1) = \{0.9271, -0.2500\}$$

$$\boldsymbol{X}_2(1) = \{-1.4325, 1.0463\}$$

$$\boldsymbol{X}_3(1) = \{-1.7474, -0.3706\}$$

Figure 8 shows the position of the estimated states of the SKF agents during estimation after re-initialization of the estimated state's dimension that falls outside the search space. An agent with '+' marking is the agent that holds the best fitness and thus named as the best-so-far solution, $\boldsymbol{X}_{\text{true}}$. Note that the first agent retains its estimation of the optimum solution from initialization to estimation because it holds the best-so-far solution for the corresponding iteration. This best-so-far solution, however, is responsible to guide the other agents to make an informed exploration or exploitation on the specific region of the search space.

The fitness evaluation, predict, measure, and estimate will be repeated until the stopping condition is met. Table 1 to Table 3 give a summary of the agents' predict, measure and estimate values from t = 1 to t = 5 with their corresponding Kalman gain, and fitness value.

The fitness trends of all the three agents from iteration 0 to 5 are shown in Figure 9. From the figure, it is apparent that agent 1 initially hold the best fitness, thus is responsible to lead the search. However, in iteration two, the agent 2 makes a better estimation, thus becoming the best-so-far solution and then leads the other agents to locate the optimal solution in iteration three. This results in a better estimation by agent 1, thus agent 1 becomes the best-so-far solution again and continues to influence other agents in the next three iterations.

Conclusions

This paper is the first tutorial on SKF algorithm that emphasizes on a numerical example for easy and intuitive explanations. This tutorial would be helpful to those who work on the fundamentals and applications of SKF as well as to students who are new to optimization research.

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).

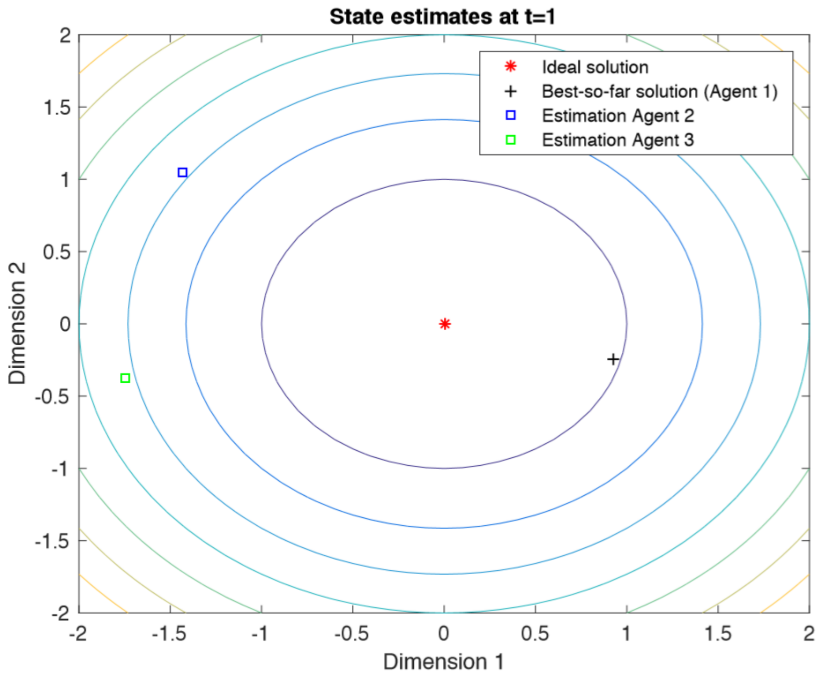

Figure 8. Estimated position of the optimum solution in the search space at t = 1.

Table 1. Summary of SKF predict, measure and estimate values from iteration 1 to 5 for agent 1.

Iteration No.	*Predict*	*Measure*	*Kalman gain*	*Estimate*	*Reinitialize*	*Fitness*
Iteration 1	{0.9271, -0.2500}	{0.9271, -0.2500}	{0.6111, 0.6992}	{0.9271, -0.2500}	-	0.9220
Iteration 2	{0.9271, -0.2500}	{0.9271, -0.2500}	{0.5824, 0.5568}	{0.9271, -0.2500}	-	0.9220
Iteration 3	{0.9271, -0.2500}	{0.0219, -0.2506}	{0.6271, 0.5956}	{0.3594, -0.2504}	-	0.1919
Iteration 4	{0.3594, -0.2504}	{0.3594, -0.2504}	{0.6507, 0.7182}	{0.3594, -0.2504}	-	0.1919
Iteration 5	{0.3594, -0.2504}	{0.3594, -0.2504}	{0.5770, 0.5972}	{0.3594, -0.2504}	-	0.1919

Table 2. Summary of SKF predict, measure and estimate values from iteration 1 to 5 for agent 2.

Iteration No.	*Predict*	*Measure*	*Kalman gain*	*Estimate*	*Reinitialize*	*Fitness*
Iteration 1	{1.7209, 0.5351}	{2.5054, 1.2498}	{0.7503, 0.7152}	{2.3095, 1.0463}	-1.4325	3.1468
Iteration 2	{-1.4325, 1.0463}	{0.0463, -0.2198}	{0.5506, 0.6512}	{-0.6183, 0.2218}	-	0.4315
Iteration 3	{-0.6183, 0.2218}	{-0.6183, 0.2218}	{0.6760, 0.4926}	{-0.6183, 0.2218}	-	0.4315
Iteration 4	{-0.6183, 0.2218}	{-0.2825, 0.6931}	{0.5873, 0.6530}	{-0.4211, 0.5295}	-	0.4577
Iteration 5	{-0.4211, 0.5295}	{-0.6553, -0.1936}	{0.5916, 0.6461}	{-0.5596, 0.0623}	-	0.3171

Table 3. Summary of SKF predict, measure and estimate values from iteration 1 to 5 for agent 3.

Iteration No.	*Predict*	*Measure*	*Kalman gain*	*Estimate*	*Reinitialize*	*Fitness*
Iteration 1	{-0.7373, -0.6772}	{-2.3303, -.2512}	{0.6341, 0.7198}	{-1.7474, -0.3706}	-	3.1906
Iteration 2	{-1.7474, -0.3706}	{-0.2385, -0.3899}	{0.7801, 0.6099}	{-0.5703, -0.3824}	-	0.4715
Iteration 3	{-0.5703, -0.3824}	{-0.5228, -0.4769}	{0.5182, 0.5550}	{-0.5457, -0.4348}	-	0.4868
Iteration 4	{-0.5457, -0.4348}	{-1.1897, -0.3334}	{0.5931, 0.6406}	{-0.9277, -0.3699}	-	0.9974
Iteration 5	{-0.9277, -0.3699}	{-0.0967, -0.4428}	{0.4980, 0.4717}	{-0.5138, -0.4043}	-	0.4275

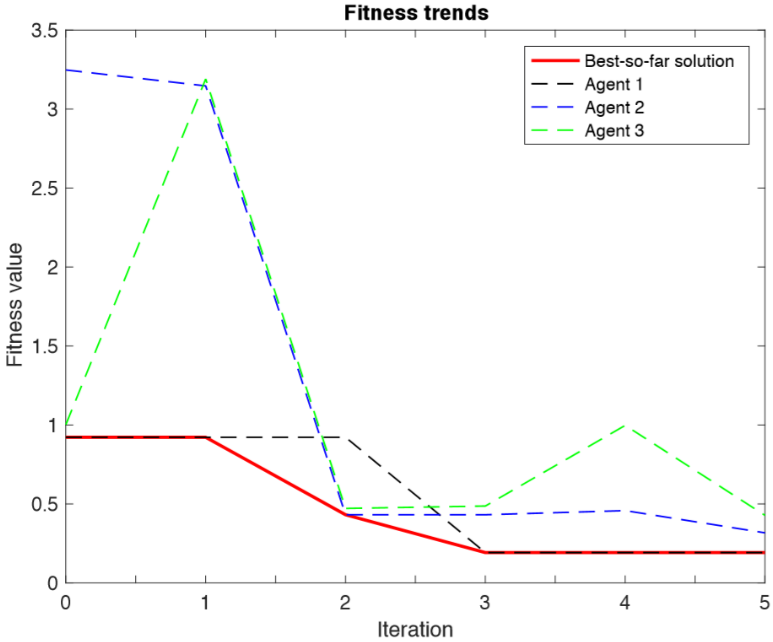

Figure 9. Fitness trends of SKF agents from iteration $t = 0$ to $t = 5$.

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.

[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 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] 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.

[34] 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.