

Received October 10, 2021, accepted October 17, 2021, date of publication October 22, 2021, date of current version October 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3122166

New Adaptive Surrogate-Based Approach Combined Swarm Optimizer Assisted Less Tuning Cost of Dynamic Production-Inventory Control System

AMIR PARNIANIFARD¹, VAHID REZAEI², SUSHANK CHAUDHARY¹, (Senior Member, IEEE), MUHAMMAD ALI IMRAN³, (Senior Member, IEEE), AND LUNCHAKORN WUTTISITTIKULKIJ¹, (Senior Member, IEEE)

¹Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand

²Department of Industrial Engineering, Yazd University, Yazd 8915818411, Iran

³School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.

Corresponding authors: Lunchakorn Wuttisittikulki (lunchakorn.w@chula.ac.th) and Amir Parnianifard (amir.p@chula.ac.th)

This work was supported by the Second Century Fund (C2F), Chulalongkorn University, Bangkok.

ABSTRACT Today, most manufacturing control systems are complex and expensive, so they are limited to employ a small number of function evaluations for optimal design. Yet, looking for optimization methods with the less-computational cost is an open issue in engineering control systems. This paper aims to propose an effective adaptive optimization approach by integrating Kriging surrogate and Particle Swarm Optimization (PSO). In this method, a novel iterative adaptive approach is utilized using two sets of training samples including initial training and adaptive sample points. The initial training points are designed by space-filling design, while the adaptive points are generated using a new jackknife resampling approach. The proposed approach can effectively convergence towards the global optimal point using a small number of function evaluations. The efficiency and applicability of the proposed algorithm are evaluated using the optimal design of the fractional-order PID (FOPID) controller for some benchmark transfer functions. Then, the introduced approach is applied for tuning the parameters and the sensitivity analysis of the FOPID controller for a dynamic production-inventory control system. The results are in good agreement with the results reported in the literature, while the proposed approach is executed with a lower computational burden.

INDEX TERMS Fractional-order PID, kriging, particle swarm optimization, production-inventory control, sensitivity analysis.

I. INTRODUCTION

A. STATEMENT OF PROBLEM AND MOTIVATIONS

Control engineering refers to the use of automatic control to make systems or processes reach the desired behavior while operating under certain constraints. This discipline has been being intensively enlarged over the past decades due to the advancement of modern technologies and the development of new systems, in particular intelligent systems [1]. Proportional-Integral-Derivative (PID) controllers with feedback control structures are widely used for industrial process

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy¹.

control. While more powerful control techniques are readily available, the PID controller is still popular due to its relative simplicity and applicability to a wide range of industrial control problems, see [2]–[5]. PID controller employs proportional, integral, and derivative gain parameters to reduce an error signal (e) to be closer to zero. The output of a PID controller equal to the control input of the plant in the time-domain is:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

where $u(t)$ is the overall control function and K_p , K_i , and K_d denote the proportional, integral, and derivative gain parameters, respectively. $e(t)$ calculates an error value as

the difference between the desired setpoint and a measured process output in the time moment t . The controller attempts to minimize the error over time by adjusting a control variable $u(t)$. The appropriate function of PID strongly depends on the allocated value for each gain of K_p , K_i , and K_d .

Currently, fractional-order controllers are being extensively used by many scientists to achieve the most robust performance of the systems [6], [7]. The main reason for choosing Fractional-Order PID (FOPID) controllers is their additional degrees of freedom that result in a better control performance [2], [8]. A generalized FOPID controller was first introduced by [9] which proposed $PI^\lambda D^\mu$ controller-involving a λ order integer and a μ order differentiator. The differential equation of a fractional order $PI^\lambda D^\mu$ controller is defined by:

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^\mu e(t) \quad (2)$$

The reliability of the FOPID controller depends on the optimal design of three gain parameters (K_i , K_p , and K_d) and two order parameters (λ , μ). There are a few software tools available for designing and tuning fractional-order controllers. Tuning a fractional PID controller is a challenging task because there are five parameters to tune. This means that there are two more parameters as compared to a classical PID controller [2], [7].

The main motivation of the current study is to develop a new method for tuning FOPID controllers when the proposed method is specified in two main specifications including i) a black-box method that does need applying dynamic mathematical expressions of a control system, and ii) a less-expensive method with the lower number of function evaluations. Here, it should also be recalled the ‘‘No Free Lunch’’ (NFL) theorem presented by [10]. This theorem has logically proved that there is no best-suited optimization method for solving all optimization problems. In other words, a particular optimization method may show very promising results on a set of problems, but the same algorithm may show poor performance on a different set of problems [11]. This theorem also motivates the current study to develop a new optimization method for tuning the FOPID controller.

B. RELATED WORKS

Different tuning methods such as the Ziegler-Nichols method, auto-tuning, self-tuning method, and optimization-based tuning for the fractional-order controller have been proposed in the literature [1], [12], [13]. However, there is still a challenging research problem since most of the integer and fractional-order PID controllers are suffering from badly-tuned parameters, being the reason why they operate improperly in non-optimal regions [14], [15]. The optimization-based tuning methods for the complex control systems are classified into mathematical methods, simulation-based methods, and surrogates [16], [17]. Investigating less computationally expensive (i.e., a smaller number of iterations or function evaluations) methods for

achieving optimal design controller has also become a main challenging topic in the control engineering practice. To overcome such computational difficulties, researchers have applied surrogate-based learning methods (e.g. polynomial regression, Kriging, and radial basis function) [18], [19]. The Kriging surrogate has been used as a widespread global approximation technique that is applied widely in control design problems [20], [21].

The best technique to adopt depends on the complexity level of the control system. Some other well-known metaheuristic methods applied in intelligent control are Grey Wolf Optimizer (GWO) [22], GA [23], [24], Particle Swarm Optimization (PSO) [24], [25], Ant-Colony Optimization (ACO) [26], [27], and Evolutionary Programming (EP) [24], Ant Lion Optimizer (ALO) [28]. The simplicity and global characteristics of such metaheuristics have been the main reasons for their extensive applications in off-line optimum control system design [29]. There are also some main modifications in the performance of mentioned optimizers applied in the engineering design, see [30]–[33].

Optimal control theory is a branch of mathematics developed to find optimal ways to control a dynamic system. One of the common control problems in engineering design applications is a production/inventory control system [34]. A block diagram representation of classical inventory and order-based production control system is schematically shown in Figure 1 [35], [36]. In the production-inventory control system, the controller aims to generate sophisticated decisions that control the order rate and the inventory level based on trade-offs between production-inventory cost and customer satisfaction [37], [38]. Order policy can be defined as PID controller systems. This kind of control for the production-inventory system is frequently used in a variety of applications [37], [39]. The application of the control theory in inventory-production systems has been illustrated in different studies, see [40]–[43].

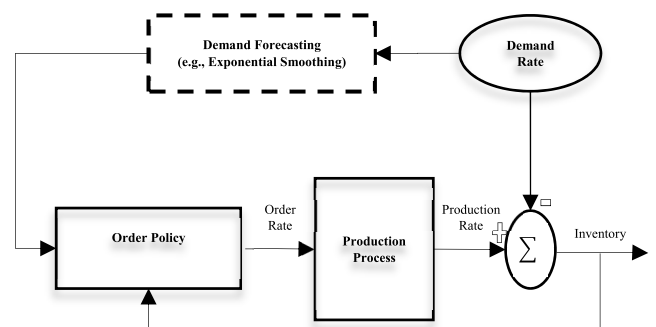


FIGURE 1. Traditional block diagram representation of a production-inventory system.

C. MAIN CONTRIBUTIONS

The main contributions of our study can be summarized as follows:

- In this paper, we introduce a new easy-to-implement algorithm for the optimal design of the

FOPID controller. This optimization approach can adaptively and effectively converge to a global optimum point. In this approach, we employ the advantages of both the Kriging surrogate and PSO metaheuristic, combined with a new framework of the jackknife leave-one-out approach.

- The proposed method is inexpensive in terms of the required number of function evaluations for the optimization procedure. This method is proper for such expensive FOPID control systems when the model is limited to a small number of function evaluations (also called simulation experiments).
- A new dynamic model for a production-inventory control system is introduced. We also define a new objective function applied in FOPID optimization. The proposed optimization algorithm is used to tune the FOPID controller for such a production-inventory control system. Computational results indicate that the proposed algorithm outperforms alternative approaches from accuracy, robustness, and performance perspectives.

Notably, as aforementioned, there are a variety of optimizers and modifications in the literature which have been developed to employ efficiently in different engineering design applications. However, the current paper limits its scope to compare the proposed algorithm with the five most common optimizers which frequently used in the tuning of integer and fractions PID controllers namely GA, ACO, PSO, GWO, and ALO. The rest of this paper is organized as follows. Section 2 provides the preliminaries required for our proposed algorithm. The algorithmic framework of the proposed approach is given in Section 3. The design procedure of the fractional PID controller for three typical benchmark transfer functions and a new framework for a dynamic production-inventory control system are presented in Section 4. Finally, this paper ends by presenting the limitations and advantages of the current study. Finally, some concluding remarks are mentioned in Section 5.

II. PRELIMINARIES

In what follows, first the basic definition of Kriging surrogate is given. Subsequently, the PSO method as a common optimizer in the control engineering practice is presented.

A. KRIGING

Since a Kriging was first introduced by Krige [44] in 1951, it has been used as a widespread global approximation technique that is applied widely in engineering problems [45]. Kriging is an interpolation method that can cover deterministic data and is highly flexible due to its ability to employ a various range of correlation functions. The higher accuracy of Kriging models comparing to the other alternatives such as response surface modeling is confirmed in the literature [46], [47]. In a Kriging model, a combination of a polynomial model and realization of a stationary point is assumed by the form of:

$$y = \sum_{j=0}^k \hat{\beta}_j f_j(X) + Z(X) + \varepsilon \quad (3)$$

The polynomial terms of $f_j(X)$ are typically first or second order of response surface approach and coefficients $\hat{\beta}_j$ are regression parameters ($j = 0, 1, \dots, k$). The term ε describes approximation error and the term $Z(X)$ represents the realization of a stochastic process which is the most time normally distributed Gaussian random process with zero mean, and variance σ^2 , and non-zero covariance. The correlation function of $Z(X)$ is defined by:

$$\text{Cov}[Z(x_i), Z(x_p)] = \sigma^2 R(x_i, x_p) \quad (4)$$

where σ^2 is the process variance and $R(x_i, x_p)$ is the correlation function that can be chosen from different functions proposed in the literature. Due to the tuning of the correlation function with sample data, the Kriging is extremely flexible to capture the nonlinear treatment of the model. Among the studies in the literature, some studies have been found which sufficiently describe Kriging metamodel methodology, see [45], [48].

B. PARTICLE SWARM OPTIMIZER

The canonical PSO algorithm was proposed by [49] and is inspired by the social behavior of swarms such as bird flocking or fish schooling. The parameters of PSO consist of the number of particles, position of the agent in the solution space, velocity, and neighborhood of particles (communication of topology). The PSO algorithm begins with initializing the population. The second step is to calculate the fitness values of each particle, followed by updating individual and global bests as the third step. Then, velocity and the position of the particles become updated (step four). The second to fourth steps are repeated until the termination condition is satisfied [50], [51]. The velocity and position of the PSO algorithm are formulated as follows [49], [50]:

$$v_{id}^{t+1} = w \cdot v_{id}^t + c_1 \cdot \text{rand}(0, 1) \cdot (p_{id}^t - x_{id}^t) + c_2 \cdot \text{rand}(0, 1) \cdot (p_{gd}^t - x_{id}^t) \quad (5)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (6)$$

where w is the inertia weight factor and v_{id}^t and x_{id}^t are particle velocity and particle position respectively. The expression d is the dimension in the search space, i is the particle index, and t is the iteration number. Expressions c_1 and c_2 represent the speeds of regulating the length when flying towards the most optimal particles of the whole swarm and the most optimal individual particle. The term p_i is the best position achieved by particle i so far and p_g is the best position found by the neighbors of particle i . The expression $\text{rand}(0, 1)$ shows the random values between 0 and 1. The exploration happens if one or two of the differences are large, i) differences between the particle's best (p_{id}^t) and previous particle's position (x_{id}^t), ii) differences between the population's all-time best (p_{gd}^t) and previous particle's position (x_{id}^t). Besides, exploitation occurs when these two values are both small. PSO has attracted wide attention in control engineering

design problems due to its algorithmic simplicity and powerful search performance [52]. However, a PSO algorithm that requires a large number of fitness evaluations before locating the global optimum is often prevented from being applied to computationally expensive real-world problems [53]. Therefore, surrogate-assisted PSO metaheuristic optimization algorithms have been focused on in the literature, see [21], [54].

C. CONVENTIONAL JACKKNIFE RESAMPLING

Jackknife was first introduced by Quenouille [55] and was named by Tukey [56]. The application of the jackknife method involves a leave-one-out strategy for the estimation of a parameter (e.g., the variance) in a dataset. In general, the jackknife resampling approach has been used to estimate surrogate prediction error and investigates new sample points to improve the accuracy of a surrogate [57]. This method is based on the leave-one-out approach, and it uses an existing set of data and is not required to re-run the expensive simulation model. Assume the initial set of training points is designed with n training points ($i = 1, 2, \dots, n$). To select a new sample point between c candidate points ($j = 1, 2, \dots, c$), and add it into the set of training points, drop one point from the current set of training points, $-i$, and construct a surrogate based on $n - 1$ remaining points. Note that to avoid extrapolating by a surrogate (e.g., Kriging, radial basis function, or neural networks), the sample points on the vertices are not dropped. The jackknife's pseudo-value for candidate j is calculated as below:

$$\tilde{y}_j^{-i} = n \times \hat{y}_j^{-0} - (n - 1) \times \hat{y}_j^{-i} \quad \text{for } i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, c \quad (7)$$

where \hat{y}_j^{-0} is the original prediction for candidate j with surrogate over the whole training sample points and \hat{y}_j^{-i} is the prediction for candidate j with surrogate over $n - i$ sample points (remove i th sample point from n set of points). The jackknife's variance \tilde{s}_j^2 is computed for candidate j by employing relevant pseudo-values:

$$\tilde{s}_j^2 = \frac{1}{(n - 1)} \sum_{i=1}^n (\tilde{y}_j^{-i} - \tilde{y}_j)^2, \quad \tilde{y}_j = \frac{1}{n} \sum_{i=1}^n \tilde{y}_j^{-i} \quad (8)$$

So, the candidate with maximum \tilde{s}_j^2 is selected as a winner and is entered in the set of current training sample points after computing its relevant true response with an original simulation. All the steps are repeated till stopping criterion is satisfied. Among the previous studies, no specific stopping criterion was found to be appealed for in almost all cases [58]. However, it can be defined based on a limitation of computational time or cost.

III. PROPOSED ALGORITHM

In this section, the proposed promising optimization algorithm using hybrid Kriging surrogate and PSO metaheuristic based on adaptive improvement using the jackknife leave-one-out technique is explained. For this purpose, we first

explain the objective function used in the proposed optimization model. Then, we sketch the algorithmic steps in the proposed approach. This approach is specified for the optimal design of the FOPID controller in complex and expensive systems.

A. OBJECTIVE FUNCTION DEFINITION

For an optimal design of a FOPID controller, a suitable objective function that represents system requirements should be defined based on some desired specifications. Some conventional output specifications in the time-domain are the overshoot (M_p), rise time (T_r), settling time (T_{ss}), and steady-state error (E_{ss}). It is worth noticing that using different performance indices makes different solutions for the optimal FOPID controllers [7], [52]. Time-domain performance criteria including IAE, ISE, and ITSE formulas are defined as follows:

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (9)$$

$$IAE = \int_0^{\infty} |e(t)| dt \quad (10)$$

$$ITSE = \int_0^{\infty} t \cdot e(t)^2 dt \quad (11)$$

These performance criteria have advantages and disadvantages. For example, one disadvantage of IAE and ISE criteria is that their minimization can result in a response with a relatively small overshoot but a long settling time [52]. Although, the ITSE performance criterion can overcome the disadvantage of the ISE and IAE criteria [59]. Here, we propose and use a new cost function using all time-domain specifications including overshoot, rise time, settling time, and steady-state error besides the ITSE performance index as below:

$$C_{\alpha}(K) = \alpha(M_p + E_{ss}) + (1 - \alpha)(T_r + T_{ss}) + ITSE \quad (12)$$

where $K = [K_p, K_i, K_d, \lambda, \mu]$ is the parameter vector and $\alpha \in [0, 1]$ is the weighting factor. The optimum selection of α depends on the designer's requirements and the characteristics of the plant under control. To reduce the rise time and settling time, α can be set to be small (close to 0). Alternatively, if α is to be set larger (close to 1), then the overshoot and steady-state error are reduced more. In contrast, if α is set to 0.5, then all the performance criteria (i.e., the overshoot, rise time, settling time, and steady-state error) will have the same importance for optimization in the objective function of the FOPID model.

B. A NEW JACKKNIFE RESAMPLING CRITERION

The conventional jackknife approach in Eq.(7) considers the prediction error to select the best points among some candidate points and adding them adaptively into the set of training sample points to increase the accuracy of approximation by the surrogate. However, the conventional formulation in Eq.(7) investigates the candidate points with a higher jackknife variance (Eq.(8)), when it doesn't consider the possibility that the selected point between candidate points is located

surrounding the global optimal point in a design space. In other words, the winner point (with higher jackknife's variance) between all the candidate points, may or may not have a lower fitness function in an optimization model under study. In this study, we aim to adaptively estimate the possible location and investigate the global optimal point in the design space inspired by the conventional formulation in Eq.(7). Accordingly, we propose a new formulation of the jackknife leave-one-out approach. Assume that the initial set of training points is designed with n training points ($i = 1, 2, \dots, n$); to select a new sample point between c candidate points ($j = 1, 2, \dots, c$), we apply a new jackknife leave-one-out formula as below:

$$L_j = n \times \left[y_{min}^{-0} - \hat{y}_j^{-0} \right] + (n - 1) \times \left[y_{min}^{-0} - \left(\frac{1}{n} \sum_{i=1}^n \tilde{y}_j^{-i} \right) \right]$$

for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, c$ (13)

where y_{min}^{-0} is the smaller value of fitness function among all the points in the set of the current training points. This value is obtained using the output of the true simulation model for each sample point in the set of training points. The expression \hat{y}_j^{-0} denotes the prediction of the j th candidate point using the surrogate that is constructed overall training points. The expression \tilde{y}_j^{-i} shows the prediction of the j th candidate point using the surrogate that is constructed over the set of training points except for i th sample point. Between all $j = 1, 2, \dots, c$ candidate points, the point with higher value L_j is selected as a winner and is added to the current set of training points. Typically, Eq.(13) investigates the point in design space with a higher possibility of improvement compared to the other candidate points. Notably, since Eq.(13) employs a surrogate to predict output for each candidate point instead of running the true simulation model, it doesn't increase the computational cost and the required number of function evaluations for the optimization procedure. Accordingly, we can evaluate a larger set of candidate points in each iteration without concerning the computational cost. In this study, we employ the PSO optimizer to produce candidate points and investigate a point with maximum L_j in Eq.(13).

C. ALGORITHMIC FRAMEWORK

In this study, we propose a new adaptive optimization algorithm by integrating the Kriging surrogate, PSO optimizer, and a new jackknife leave-one-out approach. The flowchart of the design procedure of the proposed algorithm is shown in Figure 2. Notably, this algorithm is low-cost to implement for such expensive systems when the system is limited to a small number of function evaluations required for the optimization model. Here, we specify the proposed algorithm to apply in a less expensive optimal design of the FOPID controller. In the FOPID optimization model, five decision variables including three gains K_i, K_p, K_d and two order λ, μ parameters are considered. In this model, Eq.(12) is considered as the objective function. Algorithm 1 sketches the

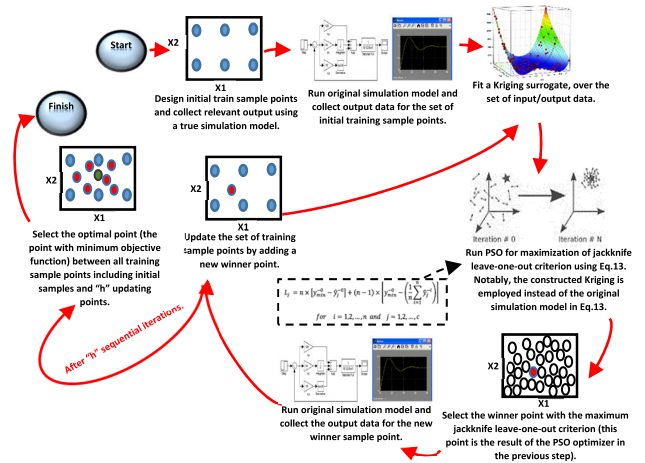


FIGURE 2. The flow diagram of the proposed algorithm.

algorithmic framework and the main steps of the proposed approach.

In this algorithm, the set of training sample points for fitting Kriging and interpolating the whole design space includes i) initial training points and ii) update points. Here, we use a common space-filling design method namely Latin hypercube sampling (LHS) to design the initial training sample points. LHS was first introduced by McKay and colleagues [60]. It is a strategy to generate random sample points while guaranteeing that all the portions of the design space are depicted. In general, for n input variables, m sample points are produced randomly in m intervals or scenarios (with equal probability). We apply a PSO metaheuristic to investigate the whole design space to select the best sample point and update the training set which is applied to reconstruct the Kriging surrogate accordingly. In the proposed algorithm, to select the update point in each sequential run, the PSO optimizer looks for a point with a larger jackknife value in Eq.(13). These new update points can adaptively improve the accuracy of Kriging interpolation. This surrogate can interpolate the whole design space particularly locations with the higher possibility to surround the optimum points. For a typical transfer function as an instance, the initial train sample points, and adaptive sample points for the FOPID tuning problem are shown in Figure 3 using the proposed algorithm.

Notably, the optimization procedure for this typical transfer function including other instances will be discussed more in Section 4. As can be seen in this example, two sets of sample points are produced through the optimization procedure including initial sample points (e.g., here the initial sample size is designed with 15 points) and the adaptive sample points that sequentially are designed and added into the set of train points to update Kriging (here also 15 adaptive points are obtained through 15 sequential iterations). The optimum point obtained for this example is shown for each combination of decision variables $[K_i, K_p, K_d, \lambda, \mu]$. It should be noted that in this figure, the size of bubbles depicts the value of the objective function. It can be seen that the adaptive

Algorithm 1 The Proposed Algorithm in Pseudocode

Input: The upper and lower bound of decision variables (i.e. five decision variables for FOPID tuning is $K = [K_p, K_i, K_d, \lambda, \mu]$), the parameter $\alpha \in [0, 1]$ in Eq.(12), and a total number of sequential iterations to update the model (h).

Output: The optimal point found by the proposed approach.
begin

- Step 1. Design initial training sample points using space-filling design methods (e.g., Latin hypercube sampling) with size n , $i = 1, 2, \dots, n$.
- Step 2. Run the original model (i.e. here we use simulation model) for each sample point and obtain the relevant true output y_i regarding the objective function (here $C_\alpha(K)$ as formulated in Eq.(12) is used as the objective function for the FOPID tuning problem).
- Step 3. Set $h = 1$ \blacktriangleright Sequential iteration's number.
- Step 4. Fit a Kriging surrogate over sets of input/output data pairs.
- Step 5. Run the PSO optimizer and select the optimizer's result as a winner point:
 - Set the maximization of a jackknife leave-one-out criterion in Eq.(13) as the fitness function for PSO.
 - Employ the constructed Kriging to obtain the predicted output (an approximation for the objective function of the optimization model) \hat{y}_j , ($j = 1, 2, \dots, c$) applied in Eq.(13).
- Step 6. Run the original simulation model and obtain the true simulation output for the selected winner sample point.
- Step 7. Update the set of input/output data pairs (set of training sample points).
- Step 8. Update Kriging surrogate over the new set of input/output data pairs.
- Step 9. Update $h = h + 1$.
- Step 9. Repeat Step 5 till Step 9 until reach the total number of sequential iterations, h (stopping criterion).
- Step 10. Select the point with minimum objective function (C_α) between all training sample points (including initial samples and h adaptive sample points) as the model's optimal point.

end

sample points with proper accuracy (lower objective function) can converge in the place that the final optimal result is located. In other words, the proposed algorithm can estimate the location of the optimum point with a small number of sample points (e.g., 30 sample points in this example) including initial sample points and adaptive (sequential) sample points. In the following section, we evaluate the applicability and effectiveness of the proposed algorithm

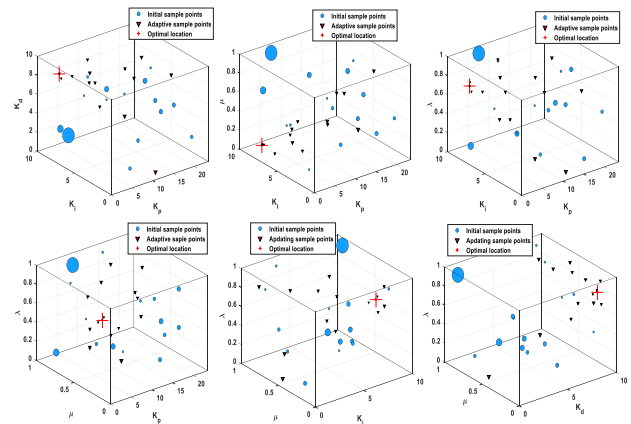


FIGURE 3. Design of train points including initial sample points and adaptive sample points using the proposed algorithm for some combinations of decision variables in FOPID tuning problem.

for the optimal design of the FOPID controller for three typical transfer functions and one practical case in a dynamic production-inventory control system.

IV. NUMERICAL EXAMPLES

A. OPTIMIZATION PROCEDURE SETUP

In this section, to verify the performance of the proposed algorithm, comparative experiments are carried out in the less-expensive (with the small number of function evaluations) optimal design of the FOPID controller for three typical transfer functions and a new dynamic structure of production-inventory control system. The proposed algorithm in this study is developed for such complex systems when the optimization model is restricted to employ a small number of function evaluations. To evaluate the performance and accuracy of the proposed algorithm in comparison with common intelligent control methods, we select five common optimizers that are widely used in the optimization of control systems including PSO [49], GWO [11], ALO [28], ACO [26], and GA [61]. For designing an optimal FOPID controller, determining the decision variables $K = [K_i, K_p, K_d, \lambda, \mu]$ associated to the minimization of the objective function $C_\alpha(K)$ (see Eq.(12)) is the main issue. Here, the proposed optimization algorithm is performed to minimize the objective function $C_\alpha(K)$. For this purpose, step response of the plant is used for computing all performance components in Eq.(12) including overshoot (M_p), rise time (T_r), settling time (T_{ss}) and steady state error (E_{ss}), and ITSE. Here, we assume that the optimization model is limited to only 30 function evaluations (simulation experiments). Moreover, we only let the proposed algorithm and other five optimizers PSO, GWO, ALO, ACO, and GA to employ only 30 function evaluations to obtain the optimal result for optimal design of FOPID controller. We simulate all control plants in Matlab®/Simulink environment. The size of the time-step is fixed at 0.01 and the time-domain (simulation time) is considered $T = 20$. The setpoint for all the problems is fixed to 1. Simulink does not have

a library for the FOPID. Therefore, the controller from the library of FOMCON: a Matlab® toolbox for fractional-order system identification and control [62] which allows for the computation of the fractional-order derivative and integration is used. It is assumed that the suitable ranges of the control parameters for all control problems are as follows:

$$K_p \in [0, 25], \quad K_i \in [0, 10], \quad K_d \in [0, 10], \quad \mu \in [0, 1] \\ \text{and } \lambda \in [0, 1].$$

It should be noted that the parameter α in Eq.(12) is set to $\alpha = 0.5$ for all control plants (in this case, all the performance criteria have the same merit in the objective function).

To study and compare the effect of randomness in the proposed algorithm with five common optimizers including PSO, GWO, ALO, ACO, and GA, we repeat each stochastic optimization method 10 times for each problem and compute the statistical measures including mean, Standard Deviation (Std), and the Signal to Noise Ratio (SNR) as the robustness criterion as below:

$$SNR = 10 \log \left[\mu^2 + \omega \sigma^2 \right] \quad (14)$$

where μ and σ show the mean and Std of optimizer obtained by repeating the optimization model (here 10 separate repetitions). Since we performed a minimization of the model's output (see the objective function in Eq.(12)) to find the optimal parameters of the FOPID controller, the formulation of SNR in Eq.(14) has the opposite sign by Taguchi formulation. Additionally, a weighting parameter ω is introduced to allow for individual emphasis on the minimization of variations. The smallest value of SNR in Eq.(14) depicts the better stochastic optimization method with higher accuracy (smaller mean of simulation output) and higher robustness (smaller standard deviation) against randomness in the model.

B. THREE TYPICAL TRANSFER FUNCTIONS

Three different control plants are extracted from [52] with transfer functions as below:

$$G_1(s) = \frac{1}{s^3 + 6s^2 + 7s} \quad (15)$$

$$G_2(s) = \frac{s^2 + 2s}{s^4 + 2s^3 + 5s^2 + s + 0.1} \quad (16)$$

$$G_3(s) = \frac{e^{-0.1s}}{s^2 + 2s} \quad (17)$$

The following process is performed to determine the optimal design of the FOPID parameters (K_i, K_p, K_d, λ , and μ) using the proposed algorithm for the mentioned control plants. First, we design a small set of sample points (here 15 sample points) using the Latin Hypercube Sampling (LHS) method (here we use the Matlab® function "lhsdesign" to construct LHS design). Then, we run the original simulation model to obtain relevant outputs (see Eq.(12)). Next, we fit a Kriging surrogate over a set of input/output data pairs (train sample points). After that, we run the

PSO optimizer to select another sample point to add to the set of train sample points. The maximization of L_j criterion in Eq.(13) is set as the fitness function for the PSO optimizer. The constructed Kriging is used to estimate the model's output used in Eq.(13). The result gained by PSO is selected as a winner point. We run the original simulation model again to obtain the true output in the winner point. The input-output data pair for the winner point is added to the existing set of training sample points. We also update the Kriging surrogate accordingly. This adaptive iteration is repeated until the stopping criterion is satisfied. In this study, we stop adaptive sampling when the model reaches 15 iterations (and accordingly 15 adaptive sample points are designed). Therefore, a total of 30 function evaluations regarding the designed sample points (initial and adaptive points) are performed through the implementation of the proposed algorithm for the optimal design of the FOPID controller. To make a fair comparison between the proposed algorithm and the other common FOPID solvers, we only let each optimizer e.g., PSO, GWO, ALO, ACO, and GA evaluate the main objective function 30 times. For this purpose, we set 3 initial populations and 10 maximum iterations for PSO, GWO, and ALO optimizers, 3 ants with 10 iterations for ACO, and a population size of 5 with 6 generations for GA. Other parameters are adjusted as follows. Min and max inertia weights, velocity clamping factor, cognitive constant (c1), and social constant (c2) for PSO is fixed to 0.7, 0.9, 1.5, 2, and 2 respectively. For the ACO optimizer, the evaporation rate and the number of nodes for each parameter are adjusted to 0.7 and 10,000 respectively. The obtained results for the plants $G_1(s)$, $G_2(s)$ and $G_3(s)$ are summarized in Tables 1, 2 and 3, respectively.

The pairwise comparison with statistical measures including mean, Std, and SNR, and the unit step responses of the control plants for the control plants $G_1(s)$, $G_2(s)$ and $G_3(s)$ are shown in Figures 4-9, respectively. It is seen that the proposed algorithm to tune the FOPID controllers outperform those of the PSO, GWO, ALO, ACO, and GA obtained controllers. In all the obtained results in figures, it can be observed that the proposed algorithm gives higher accuracy (lower cost function $C_\alpha(K)$, see Eq.(12)) comparing to the other five methods. Besides, the proposed algorithm shows more robustness comparing to the other stochastic optimizers with a smaller standard deviation for results obtained over 10 repetitions of the method. From the simulation results, it can be found that the proposed algorithm applied in the tuning of FOPID controllers produces the smooth curve for the output in conjunction with little fluctuation and small overshoot.

C. PRODUCTION-INVENTORY CONTROL SYSTEM

1) DYNAMIC OF PRODUCTION-INVENTORY CONTROL SYSTEM

A good implementation of the control theory applied to production-inventory control has been presented by [36], [63]–[65]. Here, we consider a factory producing single homogeneous goods and having a finished goods warehouse.

TABLE 1. FOPID optimal results in $G_1(s)$ using different methods for 10 repetitions ($\alpha = 0.5$).

Method	Output							
	C_α	ISE	ITSE	M_p	T_r	T_{ss}	E_{ss}	
Proposed algorithm	Mean	1.79	0.42	0.14	0.36	1.04	1.91	0.00
	Std	0.90	0.11	0.11	0.41	0.54	1.16	0.00
	Best	1.05	0.34	0.08	0.00	0.73	1.12	0.00
	Worst	4.30	0.74	0.44	0.96	2.63	5.08	0.01
PSO	Mean	2.45	0.37	0.10	2.28	0.82	1.60	0.00
	Std	3.18	0.07	0.05	6.02	0.25	0.61	0.00
	Best	1.06	0.33	0.08	0.00	0.52	1.10	0.00
	Worst	11.92	0.58	0.24	20.29	1.53	2.85	0.01
GWO	Mean	8.42	0.64	0.48	6.65	1.42	7.81	0.01
	Std	3.86	0.30	0.28	5.26	0.60	3.62	0.01
	Best	1.57	0.35	0.13	0.30	0.71	1.58	0.00
	Worst	16.10	1.32	1.09	16.77	2.65	13.48	0.02
ALO	Mean	18.19	0.67	1.19	24.56	1.05	8.40	0.01
	Std	5.78	0.30	2.29	5.74	0.61	4.51	0.02
	Best	13.10	0.40	0.19	16.42	0.57	3.74	0.00
	Worst	34.67	1.44	8.04	33.14	2.75	19.71	0.05
ACO	Mean	8.59	0.43	0.19	11.36	0.81	4.61	0.01
	Std	3.59	0.05	0.06	6.45	0.13	2.59	0.00
	Best	2.03	0.35	0.09	0.92	0.63	1.89	0.00
	Worst	15.39	0.52	0.29	21.34	1.04	9.20	0.01
GA	Mean	8.60	0.44	0.25	10.17	0.83	5.69	0.00
	Std	3.33	0.18	0.22	4.51	0.34	2.92	0.01
	Best	1.94	0.33	0.08	1.88	0.62	1.10	0.00
	Worst	12.49	0.96	0.85	16.96	1.81	10.05	0.02

TABLE 2. FOPID optimal results in $G_2(s)$ using different methods for 10 repetitions ($\alpha = 0.5$).

Method	Output							
	C_α	ISE	ITSE	M_p	T_r	T_{ss}	E_{ss}	
Proposed algorithm	Mean	2.28	0.08	0.08	2.09	0.22	2.07	0.02
	Std	0.27	0.03	0.10	0.63	0.06	0.35	0.01
	Best	1.90	0.05	0.02	1.46	0.17	1.64	0.01
	Worst	2.78	0.15	0.36	3.68	0.33	2.76	0.04
PSO	Mean	2.43	0.06	0.06	2.69	0.17	1.85	0.01
	Std	0.34	0.02	0.08	1.17	0.02	0.52	0.01
	Best	1.87	0.05	0.01	0.93	0.16	1.30	0.00
	Worst	2.83	0.11	0.28	4.18	0.23	2.88	0.04
GWO	Mean	3.03	0.08	0.05	2.61	0.24	3.10	0.01
	Std	0.82	0.02	0.04	0.71	0.05	1.60	0.01
	Best	2.03	0.05	0.01	1.58	0.17	1.57	0.01
	Worst	4.35	0.11	0.12	4.24	0.32	5.61	0.02
ALO	Mean	3.82	0.07	0.03	4.87	0.21	2.49	0.01
	Std	1.26	0.02	0.02	2.70	0.05	1.25	0.00
	Best	2.30	0.05	0.01	2.59	0.17	1.38	0.01
	Worst	6.74	0.13	0.07	11.04	0.33	4.58	0.02
ACO	Mean	4.85	0.08	0.11	5.26	0.23	3.99	0.02
	Std	2.31	0.03	0.11	3.34	0.07	2.28	0.01
	Best	2.20	0.05	0.02	1.00	0.14	1.60	0.01
	Worst	10.70	0.15	0.35	12.31	0.40	8.20	0.05
GA	Mean	3.76	0.07	0.06	5.27	0.19	1.94	0.01
	Std	1.15	0.02	0.07	2.61	0.04	0.67	0.01
	Best	2.08	0.05	0.01	1.77	0.15	1.28	0.01
	Worst	5.61	0.11	0.23	9.44	0.27	3.79	0.04

In this study inspired by [35], we design the dynamic of the production-inventory control model as below:

$$I(t) = (1 - \delta(t)) \cdot I(t - 1) + \exp(-\theta \cdot t) \cdot O(t) - D(t) \quad (18)$$

where $I(t)$ indicates the inventory level at time t , and $\delta(t)$, ($0 \leq \delta \leq 1$) shows a portion of the inventory that cannot be used in the next time step due to some reasons like damage or frustration. The optimal order policy $O(t)$ is designed by the FOPID controller, see Figure 10. The expression $D(t)$ shows the demand on time step t . In this model, we consider the failure rate of the product that is increased by the passing of time. So, we assume that the production rate follows a nonlinear exponential distribution, $\beta = \exp(-\theta \cdot t)$, ($0 \leq \theta \leq 1$). It means that in each time step, the production process only produces the portion of β , ($0 \leq \beta \leq 1$) from the order policy.

TABLE 3. FOPID optimal results in $G_3(s)$ using different methods for 10 repetitions ($\alpha = 0.5$).

Method	Output							
	C_α	ISE	ITSE	M_p	T_r	T_{ss}	E_{ss}	
Proposed algorithm	Mean	1.12	0.23	0.01	1.87	0.12	0.23	0.00
	Std	0.80	0.03	0.00	1.65	0.06	0.11	0.00
	Best	0.22	0.21	0.01	0.03	0.03	0.09	0.00
	Worst	2.40	0.30	0.02	4.48	0.26	0.44	0.00
PSO	Mean	2.80	0.25	0.02	4.42	0.13	1.01	0.00
	Std	2.18	0.03	0.01	3.97	0.10	1.06	0.00
	Best	0.07	0.21	0.01	0.00	0.03	0.09	0.00
	Worst	6.73	0.30	0.03	12.91	0.40	3.12	0.00
GWO	Mean	2.95	0.27	0.05	4.26	0.20	1.35	0.00
	Std	2.67	0.05	0.05	4.07	0.21	1.36	0.00
	Best	0.12	0.21	0.01	0.00	0.02	0.12	0.00
	Worst	8.58	0.38	0.14	13.55	0.63	3.66	0.01
ALO	Mean	5.29	0.27	0.03	9.36	0.16	1.00	0.00
	Std	5.35	0.04	0.02	10.10	0.15	0.98	0.00
	Best	0.70	0.21	0.01	0.78	0.01	0.07	0.00
	Worst	15.40	0.33	0.06	29.15	0.44	2.79	0.01
ACO	Mean	5.93	0.41	0.83	6.45	0.41	3.34	0.02
	Std	6.97	0.29	1.80	8.43	0.53	4.32	0.04
	Best	1.66	0.23	0.01	0.00	0.02	0.09	0.00
	Worst	25.75	1.07	5.90	30.90	1.73	14.76	0.14
GA	Mean	4.56	0.26	0.02	7.86	0.13	1.09	0.00
	Std	6.19	0.03	0.01	11.89	0.11	1.06	0.00
	Best	0.57	0.21	0.01	0.00	0.02	0.08	0.00
	Worst	22.40	0.32	0.05	42.58	0.39	3.04	0.00

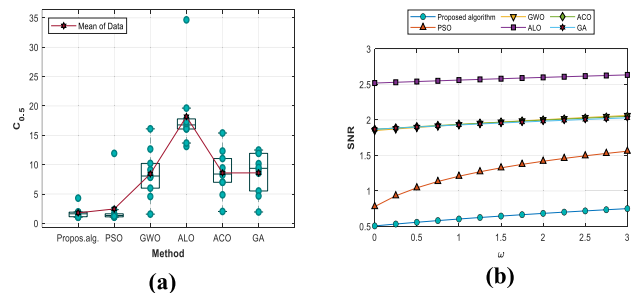


FIGURE 4. The boxplot with statistical results (a), and SNR measure by varying ω in Eq.(13) (b), for optimal design of FOPID controller in plant $G_1(s)$ using different methods. The statistical results for each optimizer obtained over 10 separate repetitions.

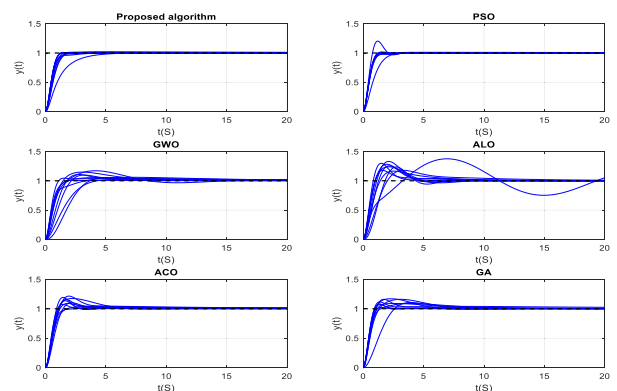


FIGURE 5. The step responses of $G_1(s)$ with FOPID controller designed by the proposed algorithm and five other common optimizers. The optimal results are obtained using 10 repetitions separately for each method.

Thus, the production rate $P(t)$ at time step t is computed by $P(t) = \beta \cdot O(t)$. The block diagram representation of the proposed dynamic production-inventory control system is shown in Figure 11.

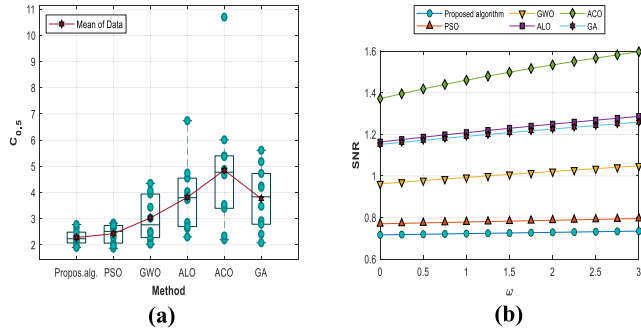


FIGURE 6. The boxplot with statistical results (a), and SNR measure by varying ω in Eq.(13) (b), for optimal design of FOPID controller in plant $G_2(s)$ using different methods. The statistical results for each optimizer obtained over 10 separate repetitions.

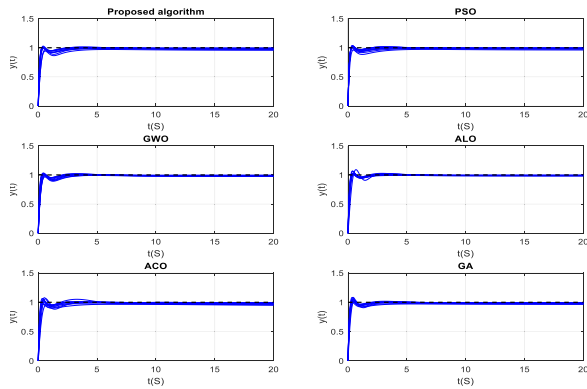


FIGURE 7. The step responses of $G_2(s)$ with FOPID controller designed by the proposed algorithm and five other common optimizers. The optimal results are obtained using 10 repetitions separately for each method.

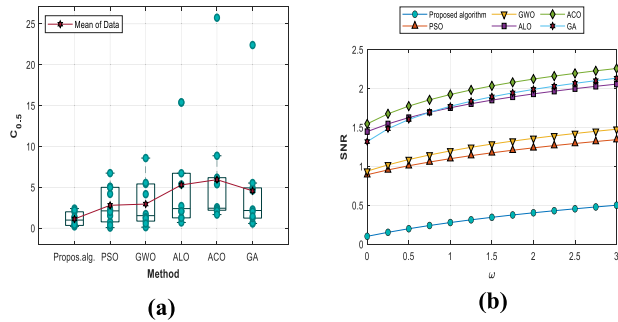


FIGURE 8. The boxplot with statistical results (a), and SNR measure by varying ω in Eq.(13) (b), for optimal design of FOPID controller in plant $G_3(s)$ using different methods. The statistical results for each optimizer obtained over 10 separate repetitions.

2) RESULTS AND DISCUSSION

To design an optimal FOPID controller based on the proposed algorithm, we implement the simulation model using Eq.(18) in the Matlab®, Simulink environment. In the current instance, we assume that the demand rate follows a nonlinear exponential distribution as $D(t) = 2\exp(0.05t)$. Besides, the failure rate is assumed $\theta = 0.02$, so production rate equals to $\beta = \exp(-0.02.t)$. We also assume that in the whole simulation period, the constant rate of 10 percent ($\delta(t) = 0.1$) from remaining inventory in time step t cannot

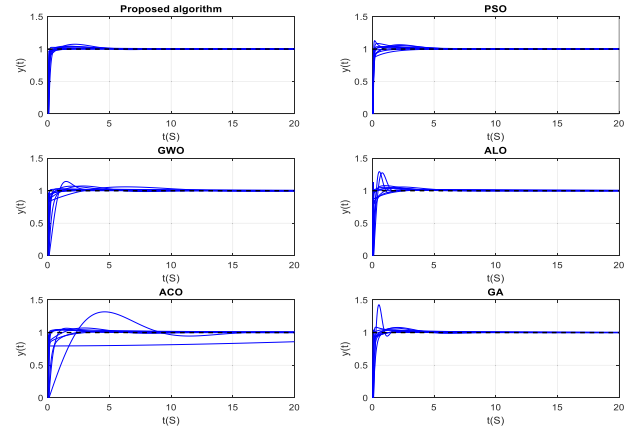


FIGURE 9. The step responses of $G_3(s)$ with FOPID controller designed by the proposed algorithm and five other common optimizers. The optimal results are obtained using 10 repetitions separately for each method.

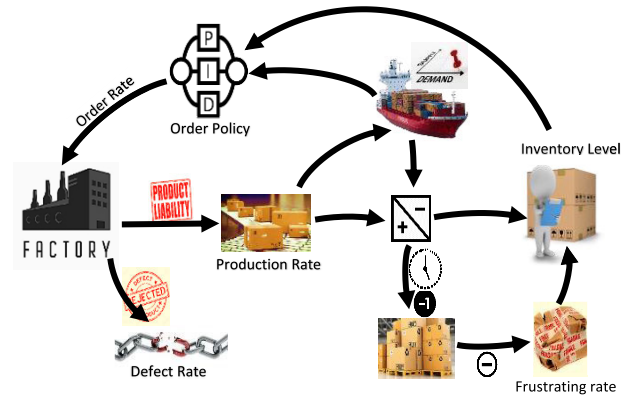


FIGURE 10. Schematic representation of the dynamic production-inventory control system. Order policy is defined based on fractional-order PID controller.

be used in the next time step $t + 1$ due to damage or frustration. The optimization process for the production-inventory control plant is done the same as what was done for three typical plants in the previous section. To determine the optimal values of the FOPID gain and order parameters ($K = [K_i, K_p, K_d, \lambda, \mu]$) by the proposed algorithm, we first design 15 sample points using the LHS method. Each sample including the vector of $K = [K_i, K_p, K_d, \lambda, \mu]$ is sent to the Simulink block and the values of five performance criteria in the time-domain (M_p, T_r, T_{ss}, E_{ss} , and ITSE) are calculated. Accordingly, the objective function $C_{0.5}(K)$ (see Eq.(12) with $\alpha = 0.5$) is computed for each sample regarding the obtained performance criteria.

Afterward, a Kriging surrogate is fitted over the set of input/output data pairs (15 samples). We employ the PSO optimizer to design a new sample point and update the set of input/output data pairs accordingly. For this purpose, we set the maximization of the proposed leave-one-out jackknife criteria regarding Eq.(13) as the fitness function in the PSO optimizer. The constructed Kriging interpolation is applied to estimate the value of the objective function required to

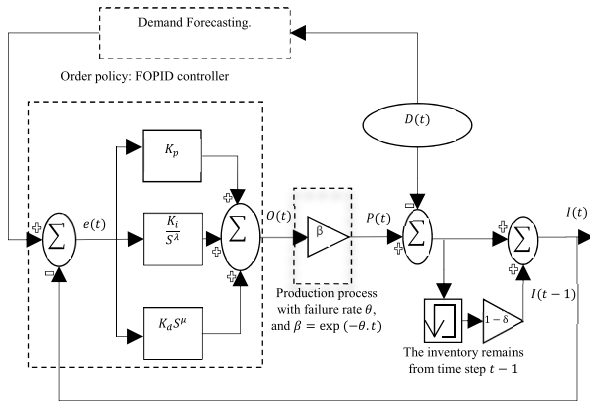


FIGURE 11. Block diagram representation of the FOPID controller in the dynamic production-inventory control system.

compute L_j (Eq.(13)) as the fitness function in PSO. As we employ Kriging instead of the original simulation model to compute the objective function, we are not worried about the computational cost with PSO optimizer in our algorithm (thus a large number of the initial population and the maximum number of iterations can be considered for the swarm optimizer). The PSO used in the proposed algorithm has the same parameter adjustment as the PSO that individually is used for direct FOPID tuning. Finally, the point with maximum L_j is selected as a winner and this new adaptive sample point is added to the set of input/output data pairs which is employed to update the Kriging accordingly. This adaptive sampling process is continued until the algorithm reaches the stopping criterion. Regarding the proposed algorithm (please see the pseudocode in “Algorithm 1”), we stop the algorithm (terminate repeating Step 5 till Step 9) when the total number of sequential iterations (h) is reached the predefined number (stopping criterion). Here for the current test functions, we stop the algorithm when 15 adaptive points are designed (thus 15 sequential iterations are derived). Finally, among all the 30 sample points (15 initial samples and 15 adaptive sample points), the best point with a lower objective function $C_{0.5}(K)$ is returned as the global best solution for K . For comparison, we also solve the FOPID tuning problem in production-inventory control system with five common optimizers of PSO, GWO, ALO, ACO, and GA. Note that to provide a fair comparison, we also let each optimizer employ 30 function evaluations to search for the optimal result. The parameter adjustment for each optimizer is the same as what was mentioned in the previous subsection for tuning three typical transfer functions. The comparative results for 10 individual repetitions of each method are summarized in Table 4.

Besides, Figure 12 shows statistical results including the boxplot, mean, standard deviation, and SNR values over the obtained results using each method.

It is seen that the FOPID tuning based on the proposed algorithm outperforms those of the FOPID tuning based on other optimizers from the accuracy (lower objective function $C_{0.5}(K)$) point of view. Besides, the proposed algorithm outperforms other optimizers in terms of robustness

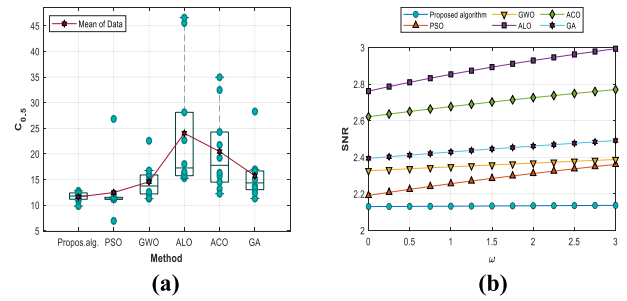


FIGURE 12. The boxplot with statistical results (a), and SNR measure by varying ω in Eq.(13) (b), for optimal design of FOPID controller in production-inventory control plant using different methods. The statistical results for each optimizer obtained over 10 separate repetitions.

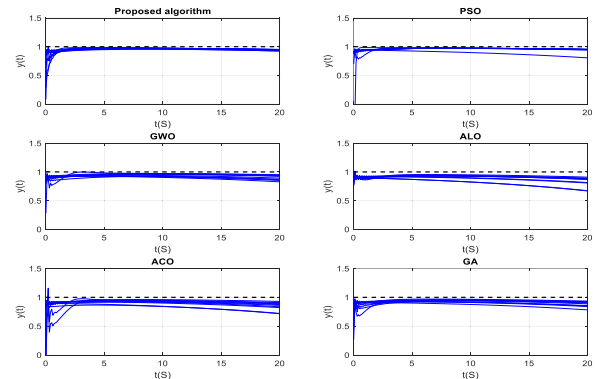


FIGURE 13. The step responses of production-inventory control plant with FOPID controller designed by the proposed algorithm and five other common optimizers. The optimal results are obtained using 10 repetitions separately for each method.

TABLE 4. FOPID optimal results in production/inventory control system using different methods for 10 repetitions ($\alpha = 0.5$).

Method	Output							
		C_α	ISE	ITSE	M_p	T_r	T_{ss}	E_{ss}
Proposed algorithm	Mean	11.62	0.06	0.39	3.46	0.66	18.28	0.06
	Std	0.84	0.03	0.14	0.85	0.50	1.71	0.01
	Best	9.81	0.03	0.25	2.12	0.06	14.33	0.05
	Worst	12.80	0.12	0.65	4.61	1.29	19.84	0.08
PSO	Mean	12.46	0.08	0.62	4.34	0.90	18.38	0.06
	Std	4.98	0.10	1.05	5.21	1.91	3.32	0.04
	Best	6.94	0.03	0.25	2.12	0.08	8.47	0.05
	Worst	26.83	0.29	3.76	19.78	6.42	19.75	0.19
GWO	Mean	14.56	0.12	1.29	6.34	0.73	19.37	0.10
	Std	3.28	0.07	0.95	3.40	1.53	1.11	0.04
	Best	11.27	0.03	0.26	2.30	0.03	16.06	0.05
	Worst	22.56	0.23	3.09	13.88	5.10	19.84	0.17
ALO	Mean	24.04	0.30	3.84	17.00	3.49	19.72	0.17
	Std	11.66	0.26	3.65	11.09	4.99	0.09	0.08
	Best	15.26	0.10	0.91	6.82	0.01	19.50	0.09
	Worst	46.60	0.80	11.03	38.93	13.08	19.79	0.33
ACO	Mean	20.46	0.28	3.09	14.31	0.97	19.30	0.16
	Std	7.54	0.21	2.74	9.98	1.65	1.01	0.07
	Best	12.23	0.06	0.57	3.47	0.01	16.68	0.07
	Worst	34.96	0.66	8.36	29.34	4.55	19.85	0.28
GA	Mean	15.74	0.17	1.78	7.48	0.96	19.36	0.12
	Std	4.56	0.11	1.33	4.26	2.28	0.61	0.05
	Best	11.30	0.06	0.46	3.76	0.01	17.77	0.07
	Worst	28.27	0.38	5.04	18.78	7.70	19.84	0.22

(smaller standard deviation) against randomness regarding different repetitions of stochastic optimization methods. On the other hand, as can be seen from Figure 12, the proposed algorithm has a smaller SNR measure for different values of parameter ω in Eq.(14) comparing to the other optimizers. Figure 13 reveals the step response of the

TABLE 5. The statistical results of sensitivity analysis for the parameter α in Eq.(12) in production-inventory control system. The results obtained over 10 repetitions using the proposed algorithm.

α	C_α		ISE		ITSE		M_p		T_r		T_{ss}		E_{ss}	
	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std
0	13.513	4.187	0.550	0.540	0.588	0.232	97.320	81.464	0.087	0.013	8.393	7.880	0.069	0.014
0.25	15.805	2.076	0.111	0.069	0.745	0.586	4.734	2.139	0.256	0.343	18.221	2.091	0.078	0.029
0.5	11.618	0.843	0.063	0.030	0.387	0.138	3.465	0.845	0.663	0.505	18.276	1.711	0.059	0.010
0.75	7.601	0.505	0.046	0.013	0.350	0.077	3.017	0.472	0.949	0.765	18.834	1.419	0.056	0.006
1	3.834	1.082	0.072	0.060	0.424	0.205	3.349	0.909	0.657	0.675	18.499	2.648	0.061	0.013

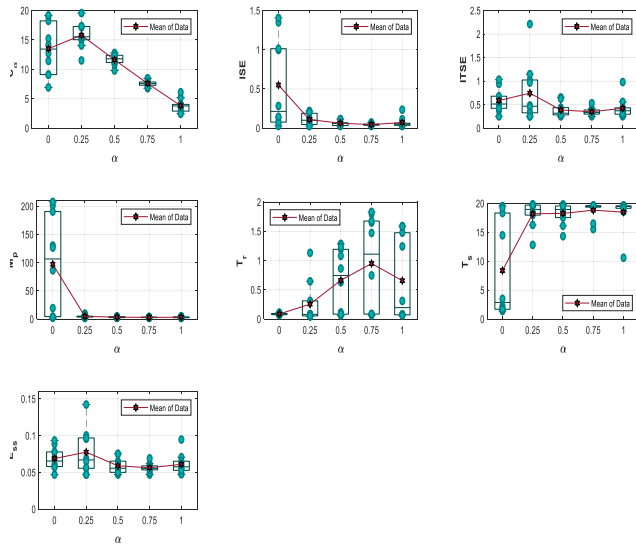


FIGURE 14. The boxplots for the results of sensitivity analysis over the parameter α in Eq.(12). Results for each value of α are obtained over 10 separate repetitions of the proposed algorithm.

motor rotation for different FOPID controllers. Obviously, the FOPID controller tuned by the proposed algorithm produces a smooth and fast output comparing to the controllers tuned by the PSO, GWO, ALO, ACO, and GA methods.

3) SENSITIVITY ANALYSIS

Finally, the sensitivity analysis is performed with different values for the parameter α in Eq.(12) to investigate the effects of the parameter α on the optimal results obtained by the proposed algorithm in the dynamic production-inventory system. To do this, the parameter α is changed from 0 to 1 with a step size of 0.25. The obtained results are given in Table 5. The statistical results for 10 repetitions of the proposed algorithm regarding each value of α are revealed in Figure 14.

As can be seen from the results, the cost function $C_\alpha(K)$ is reduced by increasing α from zero to one. It also can be seen that by increasing α from zero to one, the overshoot (M_p) and steady-state error (E_{ss}) are reduced and rise time (T_r) and settling time (T_{ss}) are increased. In general, the controller tuned by $\alpha = 0.75$ has a smaller overshoot and steady-state error, in contrast with the case $\alpha = 0$ which has a lower rise time and settling time. In addition, the best and worst ITSEs are obtained in the cases of $\alpha = 0.75$ and $\alpha = 0.25$ respectively.

V. CONCLUSION

In this paper, a new simple modified hybrid approach is proposed that integrates Kriging surrogate and particle swarm optimization to overcome the computational cost (high number of function evaluations) required in the common optimization methods in optimal design of FOPID controller. We propose a new criterion for adaptive sampling strategy inspired by a leave-one-out jackknife point of view. In the proposed algorithm, we consider increasing the accuracy of the surrogate in regions of design space that the global optimal point is located. The competitive results between the proposed algorithm and with the five most common optimizers in the tuning of control systems namely PSO, GWO, ALO, ACO, and GA show that the proposed method can provide a good balance between exploration and exploitation to the avoidance of local optima. The advantages of this methodology are its less computation burden as well as high-quality solution and speedy convergence. The results also revealed that the proposed algorithm is more robust in the stochastic environment of the optimization model. This paper has some limitations including:

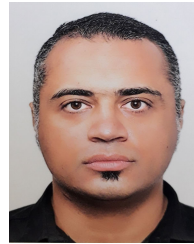
- The proposed method employs a Kriging surrogate to train the model using space-filling sampling strategies. Therefore, the approximate errors cannot be ignored when solving simulation-based optimization problems particularly with complex functions and nonlinear structures. A challenge for optimization under restricted budgets will be to find
- the right degree of approximation (smoothing factor) from a relatively limited number of samples [66].
- In this study, the application of the proposed algorithm is evaluated and compared with other optimizers for the optimal design of the FOPID controller in three benchmarks and the dynamic production-inventory control plants. However, the performance of the proposed algorithm can be examined for more challenging cases with a higher level of complexity and nonlinearity.
- Other new optimizers in the literature with modified formulation and structure compared to the five classic optimizers applied in this study for comparison of results can be employed to evaluate more the efficiency of the proposed hybrid surrogate-metaheuristic method in the tuning of the controller with a small number of function evaluations.

Therefore, future research will be devoted to overcoming the aforementioned limitations of the current study.

REFERENCES

- [1] M. Blondin, J. Sanchis, and P. Pardalos, "Control engineering from classical to intelligent control theory—An overview," in *Computational Intelligence and Optimization Methods for Control Engineering*. Cham, Switzerland: Springer, 2019, pp. 1–30.
- [2] A. Tepljakov, B. B. Alagoz, C. Yeroglu, E. Gonzalez, S. H. HosseinNia, and E. Petlenkov, "FOPID controllers and their industrial applications: A survey of recent results 1," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 25–30, 2018, doi: [10.1016/j.ifacol.2018.06.014](https://doi.org/10.1016/j.ifacol.2018.06.014).
- [3] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*, vol. 2. Durham, NC, USA: ISA Research Triangle Park, 1995.
- [4] H. O. Bansal, R. Sharma, and P. R. Shreeraman, "PID controller tuning techniques: A review," *J. Control Eng. Technol.*, vol. 2, no. 4, pp. 168–176, 2012.
- [5] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005, doi: [10.1109/TCST.2005.847331](https://doi.org/10.1109/TCST.2005.847331).
- [6] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*, 2nd ed. Research Triangle Park, NC, USA: Instrumentation, Systems and Automation Society, 1995.
- [7] P. Shah and S. Agashe, "Review of fractional PID controller," *Mechatronics*, vol. 38, pp. 29–41, Sep. 2016, doi: [10.1016/j.mechatronics.2016.06.005](https://doi.org/10.1016/j.mechatronics.2016.06.005).
- [8] R. Ranganayakulu, G. U. B. Babu, A. S. Rao, and D. S. Patle, "A comparative study of fractional order $PI\lambda/D\mu$ tuning rules for stable first order plus time delay processes," *Resource-Efficient Technol.*, vol. 2, pp. S136–S152, Dec. 2016, doi: [10.1016/j.refit.2016.11.009](https://doi.org/10.1016/j.refit.2016.11.009).
- [9] I. Podlubny, "Fractional-order systems and fractional-order controllers," *Inst. Experim. Phys., Slovak Acad. Sci., Kosice*, vol. 12, no. 3, pp. 1–18, 1994.
- [10] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [12] O. Roeva and T. Slavov, "PID controller tuning based on Metaheuristic algorithms for bioprocess control," *Biotechnol. Biotechnol. Equip.*, vol. 26, no. 5, pp. 3267–3277, Jan. 2012, doi: [10.5504/BBEQ.2012.0065](https://doi.org/10.5504/BBEQ.2012.0065).
- [13] M. J. Blondin and P. M. Pardalos, *Computational Intelligence and Optimization Methods for Control Engineering*, vol. 150. London, U.K.: Elsevier, Sep. 2019.
- [14] E. Çelik and H. Gör, "Enhanced speed control of a DC servo system using PI + DF controller tuned by stochastic fractal search technique," *J. Franklin Inst.*, vol. 356, no. 3, pp. 1333–1359, Feb. 2019, doi: [10.1016/j.jfranklin.2018.11.020](https://doi.org/10.1016/j.jfranklin.2018.11.020).
- [15] P. Van Overschee and B. De Moor, "RAPID: The end of heuristic PID tuning," *IFAC Proc. Volumes*, vol. 33, no. 4, pp. 595–600, Apr. 2000.
- [16] A. Parnianifard, M. Fakhfakh, M. Kotti, A. Zemouche, and L. Wuttisittikulkij, "Robust tuning and sensitivity analysis of stochastic integer and fractional-order PID control systems: Application of surrogate-based robust simulation-optimization," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 34, no. 2, Mar. 2021, Art. no. e2835, doi: [10.1002/jnm.2835](https://doi.org/10.1002/jnm.2835).
- [17] A. Faruq, M. F. Nor Shah, and S. S. Abdullah, "Multi-objective optimization of PID controller using Pareto-based surrogate modeling algorithm for MIMO evaporator system," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 8, no. 1, p. 556, Feb. 2018.
- [18] B. A. Asghar, "Computational intelligence and its applications in uncertainty-based design optimization," in *Bridge Optimization-Inspection and Condition Monitoring*. London, U.K.: IntechOpen, 2019.
- [19] A. Parnianifard, A. Azfanizam, M. Ariffin, and M. Ismail, "Comparative study of metamodeling and sampling design for expensive and semi-expensive simulation models under uncertainty," *Simulation*, vol. 96, no. 1, pp. 89–110, May 2019, doi: [10.1177/0037549719846988](https://doi.org/10.1177/0037549719846988).
- [20] A. Parnianifard, A. S. Azfanizam, M. K. A. Ariffin, and M. I. S. Ismail, "Kriging-assisted robust black-box simulation optimization in direct speed control of DC motor under uncertainty," *IEEE Trans. Magn.*, vol. 54, no. 7, pp. 1–10, Jul. 2018, doi: [10.1109/TMAG.2018.2829767](https://doi.org/10.1109/TMAG.2018.2829767).
- [21] A. Parnianifard, A. Zemouche, R. Chanchaoren, M. A. Imran, and L. Wuttisittikulkij, "Robust optimal design of FOPID controller for five bar linkage robot in a cyber-physical system: A new simulation-optimization approach," *PLoS ONE*, vol. 15, no. 11, Nov. 2020, Art. no. e0242613, doi: [10.1371/journal.pone.0242613](https://doi.org/10.1371/journal.pone.0242613).
- [22] R.-E. Precup, R.-C. David, and E. M. Petriu, "Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 527–534, Jan. 2017, doi: [10.1109/TIE.2016.2607698](https://doi.org/10.1109/TIE.2016.2607698).
- [23] N. Thomas and P. Poongodi, "Position control of DC motor using genetic algorithm based PID controller," *Proc. World Congr. Eng.*, vol. 2, pp. 1–3, Jul. 2009.
- [24] B. Nagaraj and N. Muruganath, "A comparative study of PID controller tuning using GA, EP, PSO and ACO," in *Proc. Int. Conf. Commun. Control Comput. Technol.*, Oct. 2010, pp. 305–313, doi: [10.1109/iccct.2010.5670571](https://doi.org/10.1109/iccct.2010.5670571).
- [25] R. G. Kanojiya and P. M. Meshram, "Optimal tuning of PI controller for speed control of DC motor drive using particle swarm optimization," in *Proc. Int. Conf. Adv. Power Convers. Energy Technol. (APCET)*, Aug. 2012, pp. 1–6, doi: [10.1109/APCET.2012.6302000](https://doi.org/10.1109/APCET.2012.6302000).
- [26] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
- [27] I. Chiha, N. Liouane, and P. Borne, "Tuning PID controller using multi-objective ant colony optimization," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, pp. 1–7, Jan. 2012, doi: [10.1155/2012/536326](https://doi.org/10.1155/2012/536326).
- [28] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: [10.1016/j.advengsoft.2015.01.010](https://doi.org/10.1016/j.advengsoft.2015.01.010).
- [29] A. Hajiloo, N. Nariman-zadeh, and A. Moieni, "Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties," *Mechatronics*, vol. 22, no. 6, pp. 788–801, Sep. 2012, doi: [10.1016/j.mechatronics.2012.04.003](https://doi.org/10.1016/j.mechatronics.2012.04.003).
- [30] H. Gao, Z. Fu, C.-M. Pun, J. Zhang, and S. Kwong, "An efficient artificial bee colony algorithm with an improved linkage identification method," *IEEE Trans. Cybern.*, early access, Mar. 23, 2020, doi: [10.1109/TCYB.2020.3026716](https://doi.org/10.1109/TCYB.2020.3026716).
- [31] H. Gao, Z. Fu, C.-M. Pun, H. Hu, and R. Lan, "A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm," *Comput. Electr. Eng.*, vol. 70, pp. 931–938, Aug. 2018.
- [32] Z.-M. Gao and J. Zhao, "An improved grey wolf optimization algorithm with variable weights," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–13, Jun. 2019.
- [33] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.
- [34] S. P. Sethi, *Optimal Control Theory—Applications to Management Science and Economics*, 3th ed. Canada: Growing Science TM, 2019.
- [35] A. Parnianifard, A. Zemouche, M. A. Imran, and L. Wuttisittikulkij, "Robust simulation-optimization of dynamic-stochastic production/inventory control system under uncertainty using computational intelligence," *Uncertain Supply Chain Manage.*, vol. 8, no. 4, pp. 633–648, 2020, doi: [10.5267/j.uscm.2020.9.002](https://doi.org/10.5267/j.uscm.2020.9.002).
- [36] A. S. White, "Management of inventory using control theory," *Int. J. Technol. Manag.*, vol. 17, nos. 7–8, pp. 847–860, 1999.
- [37] M. Ortega and L. Lin, "Control theory applications to the production-inventory problem: A review," *Int. J. Prod. Res.*, vol. 42, no. 11, pp. 2303–2322, Jun. 2004, doi: [10.1080/00207540410001666260](https://doi.org/10.1080/00207540410001666260).
- [38] A. Parnianifard, A. S. Azfanizam, M. K. A. Ariffin, and M. I. S. Ismail, "Crossing weighted uncertainty scenarios assisted distribution-free metamodel-based robust simulation optimization," *Eng. Comput.*, vol. 36, no. 1, pp. 139–150, Jan. 2020, doi: [10.1007/s00366-018-00690-0](https://doi.org/10.1007/s00366-018-00690-0).
- [39] R. W. Grubbström and J. Wikner, "Inventory trigger control policies developed in terms of control theory," *Int. J. Prod. Econ.*, vol. 45, nos. 1–3, pp. 397–406, Aug. 1996.
- [40] A. Hasani, M. Eskandarpour, and M. Fattahi, "A simulation-based optimization approach for multi-objective inventory control of perishable products in closed-loop supply chains under uncertainty," *Int. J. Adv. Oper. Manag.*, vol. 10, no. 4, pp. 324–344, 2018.
- [41] V. N. Azarskov, L. S. Zhiteckii, K. Y. Solovchuk, O. A. Sushchenko, and R. O. Lypoi, "Inventory control for a manufacturing system under uncertainty: Adaptive approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10154–10159, Jul. 2017, doi: [10.1016/j.ifacol.2017.08.1762](https://doi.org/10.1016/j.ifacol.2017.08.1762).
- [42] Y. Tao, L. H. Lee, E. P. Chew, G. Sun, and V. Charles, "Inventory control policy for a periodic review system with expediting," *Appl. Math. Model.*, vol. 49, pp. 375–393, Sep. 2017, doi: [10.1016/j.apm.2017.04.036](https://doi.org/10.1016/j.apm.2017.04.036).
- [43] V. N. Azarskov, V. I. Skurikhin, L. S. Zhiteckii, and R. O. Lypoi, "Modern control theory applied to inventory control for a manufacturing system," *IFAC Proc. Volumes*, vol. 46, no. 9, pp. 1200–1205, 2013, doi: [10.3182/20130619-3-RU-3018.00463](https://doi.org/10.3182/20130619-3-RU-3018.00463).

- [44] D. G. Krige, "A statistical approach to some mine valuation and allied problems on the Witwatersrand," *J. South. Afr. Inst. Min. Metall.*, vol. 52, no. 6, pp. 119–139, 1951.
- [45] J. P. C. Kleijnen, "Kriging metamodeling in simulation: A review," *Eur. J. Oper. Res.*, vol. 192, no. 3, pp. 707–716, 2009, doi: [10.1016/j.ejor.2007.10.013](https://doi.org/10.1016/j.ejor.2007.10.013).
- [46] Y. F. Li, S. H. Ng, M. Xie, and T. N. Goh, "A systematic comparison of metamodeling techniques for simulation optimization in decision support systems," *Appl. Soft Comput.*, vol. 10, no. 4, pp. 1257–1273, Sep. 2010, doi: [10.1016/j.asoc.2009.11.034](https://doi.org/10.1016/j.asoc.2009.11.034).
- [47] R. R. Barton, "Simulation optimization using metamodels," in *Proc. Winter Simulation Conf. (WSC)*, Dec. 2009, pp. 230–238.
- [48] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA J.*, vol. 39, no. 12, pp. 2233–2241, 2001, doi: [10.2514/3.15017](https://doi.org/10.2514/3.15017).
- [49] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1995, no. 4, pp. 1942–1948.
- [50] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PLoS ONE*, vol. 10, no. 5, May 2015, Art. no. e0122827, doi: [10.1371/journal.pone.0122827](https://doi.org/10.1371/journal.pone.0122827).
- [51] Y. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.
- [52] M. P. Aghababa, "Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm," *Soft Comput.*, vol. 20, no. 10, pp. 4055–4067, Oct. 2016, doi: [10.1007/s00500-015-1741-2](https://doi.org/10.1007/s00500-015-1741-2).
- [53] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Inf. Sci.*, vols. 454–455, pp. 59–72, Jul. 2018, doi: [10.1016/j.ins.2018.04.062](https://doi.org/10.1016/j.ins.2018.04.062).
- [54] S. Dutta, "A sequential metamodel-based method for structural optimization under uncertainty," *Structures*, vol. 26, pp. 54–65, Aug. 2020, doi: [10.1016/j.istruc.2020.04.009](https://doi.org/10.1016/j.istruc.2020.04.009).
- [55] M. H. Quenouille, "Approximate tests of correlation in time-series 3," *Math. Proc. Cambridge Phil. Soc.*, vol. 45, no. 3, pp. 483–484, Jul. 1949.
- [56] J. Tukey, "Bias and confidence in not quite large samples," *Ann. Math. Statist.*, vol. 29, no. 2, p. 614, 1958.
- [57] R. Nisbet, J. Elder, and G. Miner, *Handbook of Statistical Analysis and Data Mining*, 2nd ed. New York, NY, USA: Academic, 2017.
- [58] W. C. M. Van Beers and J. P. C. Kleijnen, "Kriging interpolation in simulation: A survey," in *Proc. Winter Simulation Conf.*, Dec. 2004, pp. 107–115, doi: [10.1109/WSC.2004.1371308](https://doi.org/10.1109/WSC.2004.1371308).
- [59] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, Jun. 2004.
- [60] M. D. Mckay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [61] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
- [62] A. Tepljakov, E. Petlenkov, and J. Belikov, "FOMCON: A MATLAB Toolbox for fractional-order system identification and control," *Int. J. Microelectron. Comput. Sci.*, vol. 2, no. 2, pp. 51–62, 2011.
- [63] J. Dejonckheere, S. M. Disney, M. R. Lambrecht, and D. R. Towill, "Measuring and avoiding the bullwhip effect: A control theoretic approach," *Eur. J. Oper. Res.*, vol. 147, no. 3, pp. 567–590, 2003, doi: [10.1016/S0377-2217\(02\)00369-7](https://doi.org/10.1016/S0377-2217(02)00369-7).
- [64] J. D. Schwartz and D. E. Rivera, "A process control approach to tactical inventory management in production-inventory systems," *Int. J. Prod. Econ.*, vol. 125, no. 1, pp. 111–124, May 2010, doi: [10.1016/j.ijpe.2010.01.011](https://doi.org/10.1016/j.ijpe.2010.01.011).
- [65] J. Wikner, *Dynamic Modelling and Analysis of Information Flows in Production-Inventory and Supply Chain Systems*. Linköping, Sweden: Profil, 1994.
- [66] S. Bagheri, W. Konen, M. Emmerich, and T. Bäck, "Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets," *Appl. Soft Comput.*, vol. 61, pp. 377–393, Dec. 2017, doi: [10.1016/j.asoc.2017.07.060](https://doi.org/10.1016/j.asoc.2017.07.060).



AMIR PARNIANIFARD received the Ph.D. degree from the Faculty of Mechanical and Manufacturing Engineering, University Putra Malaysia, in January 2019. He started his current work as a Research Assistant (Postdoctoral Fellow) with the Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand, from February 2020. He is also a PMP®certified from the Project Management Institute (PMI), USA. His research interests include computational intelligence using surrogate-based simulation-optimization with an approach to the analysis of computer experiments, adaptive improvement, robust black-box optimization, and its application in practical engineering design problems under uncertainty, such as robot manipulators, control systems, and wireless communications.



VAHID REZAIIE received the B.S. degree in industrial management from Sheikh Bahaei University, Iran, in 2014, and the M.S. degree in industrial engineering from Yazd University, Iran, in 2017. His research interests include machine learning, data mining, text mining, speech emotion recognition (SER), and image processing.



SUSHANK CHAUDHARY (Senior Member, IEEE) received the Ph.D. degree from Universiti Utara Malaysia, Sintok, Malaysia. From 2018 to 2020, he was associated with the Fujian Institute of Research on Structure and Matter, Chinese Academy of Sciences, as a Postdoctoral Researcher. He is currently associated with Chulalongkorn University, Bangkok, Thailand. He has published more than 65 research papers in SCI, and Scopus international journals and international conferences. His main research interests include autonomous vehicles, photonic radars, spatial division multiplexing, free space optics, and visible light communication systems. In 2019, he was awarded the Elite Talent Award from Jinjiang Government, China. He has delivered keynote speeches in Technology Transfer Summit, Jinan, China, and IEEE International Conference on Signal Processing and Integrated Networks, SPIN 2020 held at Noida, India.



for the IEEE COMMUNICATIONS LETTERS, IEEE ACCESS, and the *IET Communications* journals.



LUNCHAKORN WUTTISITTIKULKUJ (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from Chulalongkorn University, Bangkok, Thailand, in 1990, and the M.Sc. and Ph.D. degrees in telecommunications from the University of Essex, U.K., in 1992 and 1997, respectively. In 1995, he was involved with the European Commission Funded Project, such as Multiwavelength Transport Network (MWTN) responsible for network design and dimensioning. In 1997, he joined the Faculty of Engineering, Chulalongkorn University, where he is currently an Associate Professor with the Department of Electrical Engineering. He has authored 13 books and over 50 research articles. His research interest includes broadband wireless access networks.