1. Introduction

Nowadays, simulation-optimization methods based on DACE have been commonly interested in displaying the performance of the complex system [1]. The essential benefit of DACE method is its ability to cover complex processes in order to avoid intensive computational complexity which might squander time and resource for estimating model's parameters. There are different ways for designing of systems optimally via a mathematical model, simulation model, or meta-model. Two difficulties which arise from existing methods are computational complexity to find an optimal design and time-
Design and analysis of computer experiments using polynomial regression and Latin hypercube... 

consuming due to a large number of model's iterations [2]. Notably, choosing the best way depends on the amount of complexity in the system; see Fig. 1.

![Fig. 1. Different ways for optimizing a control system based on mathematical, simulation, or metamodel.](image)

However, different methods have been proposed in literature for designing of PID controllers optimally (e.g., evolutionary methods and heuristics algorithms); see [3]. Yet, investigating less computationally expensive methods less computationally for tuning of PID controller has been a main challenging issue in the world of control systems [4]. If the complex system not costly worth, it to be derived physically for tuning of PID controller; computer experiments through simulation model can be performed to extract approximation model instead of the real model. In such a case, metamodels (also called surrogate models) can appropriately be used to reduce the computational complexity in designing of PID controller [5]. There are several metamodels in literature applied in engineering optimization that are polynomial regression, Kriging, and Radial Basis Function (RBF) commonly [6, 7]. Undoubtedly, the most common and easy use of metamodel in simulation-optimization is polynomial regression (also called response surface methodology) that can successfully be applied in various types engineering problems [8]. However, the tuning of PID controller applying polynomial regression has been attended in some studies; see [9-12].

Computer experiments are conducted as main supplementary of meta-model based robust simulation optimization. The strength of metamodel like polynomial regression strongly depends on appropriate designing of relevant sample points which used to fit model [13]. Among different approaches for designing computer experiments, LHS has become most well-known than others [14]. LHS can cover whole design space that leads to space filling experimental design.

This paper aims to combine usage of LHS and polynomial regression for optimal design of PID controller. The proposed metamodeling approach tries to use at least number of simulation running to approximate original simulation model. So, in such a case by employing metamodel, the grid search approach can be derived in order to perform a global search over whole design space [15]. The proposed methodology in this paper is described as follows: This model reduces the computational complexity with better performance compared with traditional Zeigler-Nichols method and Taguchi-GRA. Different performance criterias that include Integral-Square-Error (ISE), Integral-Absolute-Error (IAE), Integral-Time-Square-Error (ITSE), and Integral-Time-Absolute-Error (ITAE) are compared in the same model. Note that, the proposed procedure has multi-disciplinary application and can be used...
generally for tuning of PID controllers in most types of control systems. This paper focuses on the optimal designing of PID controller for linear speed control of DC motor. DC motors have been widely employed as actuating sources in industrial application. DC motor systems with their electromechanical components are widely employed in various industrial applications due to their advantages such as easy speed control, position control, and wide adjustability interval. The new method is compared to some other types of approaches in tuning of PID controller such as metaheuristic methods [16], genetic algorithms [17], particle swarm optimization [18], fuzzy technique [19], and Taguchi method [20]. The application of metamodels (e.g. polynomial regression) has been less attended in case of complex simulations particularly using DACE.

The remainder of this paper is organized as follows: A basis of Metamodel-Based Simulation Optimization (MBSO) including polynomial regression and LHS is briefly explained in Section 2. Section 3 contains a description of PID controller for speed control of DC motor. In Section 4, the proposed MBSO algorithm to perform optimization model sketched out parallel to numerical case study in optimal design of PID controller for speed control of DC motor. Note that, the proposed polynomial regression-based optimization model consists of designing the computer experiments with LHS method, fitting polynomial regression metamodel, validating metamodel, and applying the grid search over whole design space. Finally, Section 5 is the conclusion section of this paper.

2. Metamodel Based Simulation Optimization (MBSO)

The system behavior is evaluated by running a simulation model for a fixed period of time which called computer experiments (also called simulation experiments). Metamodel can be fitted over I/O data which resulted in simulation experiments. Here, we use polynomial regression as the most common metamodel to approximate simulation and fit I/O data. The most advantages of polynomial regression are compared with other metamodels like Kriging and RBF [8]:

- The application of polynomial regression has been confirmed by large number of case studies in different engineering problems.
- Polynomial is supported by appropriate commercial software.
- Gives more information about model’s parameters such as main effect of input variables and interaction between variables.
- Handling sensitivity analysis of model’s parameters is easy.
- Supports stochastic simulation in uncertain and noisy environment.

However, in order to guarantee the validity of metamodel, the design of training sample points needs to be done over whole design space appropriately. The traditional design of experiment method for polynomial regression (e.g. Central Composite Design (CCD), Box-Behnken, and so on) [21] is not suitable for deterministic simulation, because with the same input combination by any number of re-running the model, the fixed output is gained. Moreover, here LHS as the modern method for designing of simulation experiments is used instead of the traditional methods.

2.1. Latin Hypercube Sampling (LHS)

LHS was first introduced by [22]; it is a strategy to generate random sample points, while guarantee of all portions of the design space are depicted. Generally, LHS is intended to develop results in SO. LHS has been commonly defined for designing computer experiments based on space filling concept [13]. In general, for \( n \) input variables, \( m \) sample points are produced randomly into \( m \) intervals or scenarios.
(with equal probability). For the particular instance, the LHS design for \( m = 4, n = 2 \) is shown in Fig. 2.

![Fig. 2. An example for LHS design with two input factors and four intervals.](image)

The LHS strategy proceeds as follows:

- In LHS, each input range divided into \( m \) subranges (integer) with equal probability magnitudes and numbered from 1 to \( m \). In general, the number of \( m \) is larger than total sample points in CCD.
- In the second step, LHS permutes all \( m \) intervals by random value between lower and upper bounds relevant to each interval, since each integers \( 1, 2, \ldots, m \) appears exactly once in each row and each column of the design space. Note that, within each cell of design, the exact input value can be sampled by any distribution, e.g. uniform, Gaussian, etc.

In this context for our proposed method, the LHS is performed three times to produce the needed sample points [15] by different usage that includes training sample points; validating sample points; search sample points. Training sample points are run by the original simulation model and obtained data are used for fitting metamodel. Validating sample points are run with both original simulation model and metamodel, and the extracted data employed for metamodel validating. Search sample points are run just by metamodel and the extracted data are applied in the global search to find optimal point.

### 2.2. Polynomial Regression

Polynomial regression is a collection of statistical and mathematical techniques useful for developing, improving, and optimizing the process. The functional applicability of polynomial regression in literature can approximate the relationship between design (dependent) variables and single or multi-response (independent variables), investigate and determine the best operating condition for the process by finding the best levels of design region which can satisfy operating limits, and implement robustness in the response(s) of the process by designing the process robust against uncertainty. Commonly, the main motivation of polynomial approximation for true response function is based on Taylor series expansion around a set of design points [21]. The general overview of the first-order response surface model is shown as:

\[
y = f(X) = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i + \varepsilon
\]  

(1)

where \( k \) is the number of design variables. Most times, the curvature of response surface is stronger than the first model that can approximate it even with the interaction terms. So, a second-order (quadratic) model can be employed:
\[ y = f(X) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{i' < i}^{k} \beta_{ii'} x_i x_{i'} + \varepsilon \]  

(2)

where \( \beta_0, \beta_i, \beta_{ii} \) and \( \beta_{ii'} \) are unknown regression coefficients and the term \( \varepsilon \) is the usual random error (noise) component; see [21].

3. PID Controller for Linear Speed Control of DC Motor

A DC motor is constructed with two sub-systems. The first sub-system is electrical part and consists of armature inductance, armature resistance, and the magnetic flux of the stator. The mechanical motor is the second sub-system that consists of the inertia of the motor and a load. Changing in the motor speed is due to electromagnetic moment made by the amplifier current, load, and the friction of the motor.

Fig. 3 shows the simplified electrical circuit of DC motor and the mechanical model of a rotor.

The dynamic behavior of DC motor based on the mathematical model is expressed by:

\[ J \frac{d\omega(t)}{dt} + B \omega(t) = T_m(t) \]  

(3)

\[ L \frac{d}{dt} i_a(t) + Ri_a(t) = v_a(t) - v_b(t) \]  

(4)

\[ v_b = K_e \omega(t) \]  

(5)

\[ T_m(t) = K_t i_a(t) \]  

(6)

We modeled DC motor in MATLAB® Simulink toolbox by load parameters which are depicted in Table 1.
Here, we use the same example with [20] to show the analytical comparison between our proposed method, classical tuning by Ziegler-Nichols method, and the method proposed by [20] which combined Taguchi approach and gray relational analysis (Taguchi-GRA). The PID controller employs three terms of proportional, integral and derivative to reduce an error signal \( e \) to be more close to zero. The output of a PID controller, equal to the control input to the plant, in the time-domain is as follows:

\[
    u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt}
\]

(7)

The appropriate function of PID strongly depends on allocated magnitude to each gain of \( K_p, K_i, \) and \( K_d \). In order to evaluate the performance of PID controller, four common creations are used. Note that all creations are associated with all time-response parameters included rise time, steady state error, settling time, and overshoot [23]. The mathematically features of performance creations in time domain are defined as below:

\[
    \text{Min } ISE = \int_0^\infty e(t)^2 dt
\]

(8)

\[
    \text{Min } IAE = \int_0^\infty |e(t)| dt
\]

(9)

\[
    \text{Min } ITSE = \int_0^\infty t. e(t)^2 dt
\]

(10)

\[
    \text{Min } ITAE = \int_0^\infty t. |e(t)| dt
\]

(11)

In above equations, \( e(t) = (y(t) - S_p) \), where \( y(t) \) is response (output) in time \( t \) and \( S_p \) is desired set point for response of model (i.e. here the rotational speed is defined as an output in the DC motor system).

4. MBSO Algorithm

Inspired by [1, 24], the MBSO algorithm is sketched out in some steps as follows. Here, we extend this approach for the tuning of PID controller for linear speed control of DC motor (i.e. the magnitudes of load’s parameters are stated in Table 1).

### Table 1. Load parameters in DC motor system.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Title</th>
<th>value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_a )</td>
<td>Input voltage</td>
<td>---</td>
<td>Voltage</td>
</tr>
<tr>
<td>( R )</td>
<td>Armature resistance</td>
<td>2.45</td>
<td>Ohm</td>
</tr>
<tr>
<td>( L )</td>
<td>Armature inductance</td>
<td>0.035</td>
<td>mH</td>
</tr>
<tr>
<td>( J )</td>
<td>Moment of inertia</td>
<td>0.022</td>
<td>Kgm²</td>
</tr>
<tr>
<td>( K_t )</td>
<td>Motor torque constant</td>
<td>0.015</td>
<td>Nm/Amp</td>
</tr>
<tr>
<td>( K_e )</td>
<td>Back emf constant (electromotive force constant)</td>
<td>1.2</td>
<td>Vrad/s</td>
</tr>
<tr>
<td>( B )</td>
<td>Motor viscous friction</td>
<td>0.0005</td>
<td>N.m.s</td>
</tr>
<tr>
<td>( \omega )</td>
<td>Angular velocity of the shaft</td>
<td>---</td>
<td>Rad/s</td>
</tr>
</tbody>
</table>
Step 1. Constructing simulation experiments with LSH method

If $k$ is number of input variables, in order to construct metamodel, the size of training sample points ($n_t$) needs to be at least $n_t = 10k$. We employ MATLAB®/DACE toolbox to design training samples points with LHS method. Here the input variables are $K_p$, $K_i$, and $K_d$, so $k = 3$; we need to design 30 simulation experiments ($n_t = 30$). In order to define design space for input variables, we use the traditional Ziegler-Nichols method that gives the PID gain parameters as $K_p = 1.5865$, $K_i = 1.0018$, $K_d \in [0.6281]$ (i.e. traditional Ziegler-Nichols tuning is done by MATLAB®/SISI design tool ). So, according to magnitude for each gain parameters, design ranges for $K_p$, $K_i$, and $K_d$ are assumed as bellows:

$$K_p \in [1, 2], \quad K_i \in [0.8, 1.2], \quad K_d \in [0.2, 1]$$

Fig. 4 illustrates the schematic overview of designed sample points for each two crossed input variables.

![Fig. 4. Schematic overview of designed simulation experiments obtained by LHS method.](image)

Step 2. Running simulation and collect I/O data

For each input computation, we run the simulation model and compute the relevant output. Total running of simulation is $n_t$ that in our case is 30 runs. Regarding Eqs. (8) to (11), four different outputs (responses) are defined for the model. In order to compare all performance creations in the same time domain, we evaluate the performance of PID controller with all creations ISE, IAE, ITSE, and ITAE in $0 \leq t \leq 30$ second. Table 2 represents I/O data according to each performance criteria.

Step 3. Constructing polynomial regression

We construct one polynomial regression as a metamodel over obtained I/O data that are collected from running of simulation experiments. We use Design Expert (V.10) software to fit polynomial regression. The analysis of variation among data suggests a cubic model for all four performance criterions (ISE, IAE, ITSE, and ITAE). The mathematical terms of all four cubic polynomial regressions are represented in Table 3. Fig. 5 illustrates the 3D plot surface of all four metamodels based on two gain parameters when the third gain parameter is limited in the center of relevant design range.
For this purpose, $y_{n+1} - y_n \leq 3 - y = y$, $(\ldots)$

Table 2. I/O data based on training sample points and each model’s performance.

<table>
<thead>
<tr>
<th>s</th>
<th>Design variables $K_p$</th>
<th>Performance output ITAE</th>
<th>ITSE</th>
<th>ISE</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.06 0.92 0.82</td>
<td>858.26</td>
<td>167.04</td>
<td>133.44</td>
<td>264.10</td>
</tr>
<tr>
<td>2</td>
<td>1.33 0.82 0.87</td>
<td>684.30</td>
<td>139.21</td>
<td>124.82</td>
<td>241.23</td>
</tr>
<tr>
<td>3</td>
<td>1.10 0.81 0.23</td>
<td>613.60</td>
<td>132.54</td>
<td>123.75</td>
<td>233.59</td>
</tr>
<tr>
<td>4</td>
<td>1.16 1.18 0.73</td>
<td>759.20</td>
<td>140.66</td>
<td>117.88</td>
<td>241.83</td>
</tr>
<tr>
<td>5</td>
<td>1.82 1.09 0.60</td>
<td>417.64</td>
<td>79.15</td>
<td>92.75</td>
<td>183.76</td>
</tr>
<tr>
<td>6</td>
<td>1.77 0.88 0.46</td>
<td>393.35</td>
<td>85.20</td>
<td>98.03</td>
<td>187.18</td>
</tr>
<tr>
<td>7</td>
<td>1.48 1.13 0.60</td>
<td>547.59</td>
<td>102.08</td>
<td>103.55</td>
<td>208.07</td>
</tr>
<tr>
<td>8</td>
<td>1.98 0.94 0.88</td>
<td>416.85</td>
<td>83.61</td>
<td>96.25</td>
<td>187.51</td>
</tr>
<tr>
<td>9</td>
<td>1.70 0.99 0.74</td>
<td>486.06</td>
<td>94.11</td>
<td>101.56</td>
<td>199.77</td>
</tr>
<tr>
<td>10</td>
<td>1.10 1.10 0.54</td>
<td>732.99</td>
<td>138.81</td>
<td>119.44</td>
<td>240.65</td>
</tr>
<tr>
<td>11</td>
<td>1.24 1.02 0.37</td>
<td>590.73</td>
<td>114.85</td>
<td>111.55</td>
<td>219.77</td>
</tr>
<tr>
<td>12</td>
<td>1.93 0.96 0.78</td>
<td>415.73</td>
<td>82.53</td>
<td>95.58</td>
<td>186.59</td>
</tr>
<tr>
<td>13</td>
<td>1.40 0.93 0.42</td>
<td>523.35</td>
<td>105.75</td>
<td>108.77</td>
<td>210.54</td>
</tr>
<tr>
<td>14</td>
<td>1.35 1.19 0.50</td>
<td>579.64</td>
<td>107.53</td>
<td>105.14</td>
<td>212.91</td>
</tr>
<tr>
<td>15</td>
<td>1.56 1.08 0.55</td>
<td>497.08</td>
<td>94.25</td>
<td>100.73</td>
<td>200.09</td>
</tr>
<tr>
<td>16</td>
<td>1.50 0.90 0.20</td>
<td>425.20</td>
<td>91.24</td>
<td>101.97</td>
<td>194.08</td>
</tr>
<tr>
<td>17</td>
<td>1.85 0.98 1.00</td>
<td>490.80</td>
<td>93.90</td>
<td>101.18</td>
<td>199.91</td>
</tr>
<tr>
<td>18</td>
<td>1.96 0.83 0.35</td>
<td>316.03</td>
<td>76.47</td>
<td>92.17</td>
<td>174.32</td>
</tr>
<tr>
<td>19</td>
<td>1.59 1.15 0.32</td>
<td>432.64</td>
<td>81.83</td>
<td>93.90</td>
<td>186.56</td>
</tr>
<tr>
<td>20</td>
<td>1.77 0.89 0.68</td>
<td>442.67</td>
<td>91.98</td>
<td>101.55</td>
<td>195.69</td>
</tr>
<tr>
<td>21</td>
<td>1.72 1.00 0.97</td>
<td>533.77</td>
<td>100.66</td>
<td>104.14</td>
<td>207.12</td>
</tr>
<tr>
<td>22</td>
<td>1.20 1.04 0.91</td>
<td>792.84</td>
<td>149.00</td>
<td>123.62</td>
<td>249.79</td>
</tr>
<tr>
<td>23</td>
<td>1.03 1.05 0.93</td>
<td>935.39</td>
<td>176.63</td>
<td>132.94</td>
<td>270.42</td>
</tr>
<tr>
<td>24</td>
<td>1.63 1.06 0.48</td>
<td>452.72</td>
<td>87.11</td>
<td>97.54</td>
<td>192.29</td>
</tr>
<tr>
<td>25</td>
<td>1.66 0.85 0.80</td>
<td>507.51</td>
<td>104.89</td>
<td>108.48</td>
<td>209.13</td>
</tr>
<tr>
<td>26</td>
<td>1.43 0.95 0.39</td>
<td>504.65</td>
<td>101.55</td>
<td>106.50</td>
<td>206.44</td>
</tr>
<tr>
<td>27</td>
<td>1.46 1.14 0.26</td>
<td>465.71</td>
<td>88.61</td>
<td>97.63</td>
<td>193.78</td>
</tr>
<tr>
<td>28</td>
<td>1.18 1.12 0.63</td>
<td>712.13</td>
<td>133.56</td>
<td>116.79</td>
<td>236.36</td>
</tr>
<tr>
<td>29</td>
<td>1.29 0.86 0.28</td>
<td>534.49</td>
<td>113.17</td>
<td>113.69</td>
<td>216.50</td>
</tr>
<tr>
<td>30</td>
<td>1.87 1.17 0.66</td>
<td>419.56</td>
<td>77.24</td>
<td>90.74</td>
<td>182.02</td>
</tr>
</tbody>
</table>

Step 4. Validating metamodels with scaled residuals method

There are different approaches for validating metamodels [1]. Here, we check validation of metamodels by scaled residuals error, due to the fact that scaled residuals often give more information than other ordinary least square residuals. For this purpose, first, design $n_v$ simulation experiments (validation sample points) via LHS, then run simulation and calculate relevant outputs ($y_s$) for each points ($s = 1, 2, ..., n_v$). So, for each input combination, compute standardized residuals as below:

$$D_s = \frac{y_s - \hat{y}_s}{\sqrt{\frac{1}{T} \sum_{s=1}^{T} (y_s - \hat{y}_s)^2}}$$

where $\hat{y}_s$ is predicted output for $s$ – th input combination by metamodel that fits over training sample points. While most of the standardized residuals should be in the interval $-3 \leq D_s \leq 3$ and any observation outside of this interval (outlier) is potentially unacceptable with respect to its observed simulation output. In the current instance, we produce 20 validate sample points. As can be seen from the scatterplots for results of validation in Fig. 6, we can accept all four metamodels because all the standardized results in each metamodel are being in $[-3, +3]$. 

![Design and analysis of computer experiments using polynomial regression and latin hypercube...](Image 462x799 to 530x816)
Step 5. Investigate optimal solution

Search methods with less computational complexity is one of the main advantages of employing metamodels instead of an expensive original model. In such using metamodel, the common approach to searching optimal solution is the “grid search” method. In this approach, the whole design space is studied through evaluating of model components over equally spaced grid points [15, 25]. The grid points can be replaced by random points, which spread over whole design space (Monte Carlo random search) or some common space filling methods (e.g. LHS method). Here, the LHS method is used to produce search sample points. In this searching method, there is a chance to find the global optimum solution because the searching is not bounded to local “valley”. For the current case, we produce 10,000 sample points over whole design space by LHS method. Among all sample points, the optimum point is investigated when according to each metamodel, the objective is the minimization of ITAE, ITSE, ISE, and IAE. Here, the grid search method has computed the same optimum point for ITSE, ISE, and IAE. Table 4 represents the system performance, which compares two optimum points obtained by metamodels over ITAE and over ITSE, ISE, and IAE (give the same result), and by Taguchi-GRA result and tradition Ziegler-Nichols method.

Table 3. Cubic equations of polynomial regression for all four performance creations.

<table>
<thead>
<tr>
<th>Term</th>
<th>ITAE</th>
<th>ITSE</th>
<th>ISE</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1897.87</td>
<td>594.73</td>
<td>325.54</td>
<td>530.68</td>
<td>Constant</td>
</tr>
<tr>
<td>-1881.72</td>
<td>-382.40</td>
<td>-142.77</td>
<td>-251.47</td>
<td>* Kp</td>
</tr>
<tr>
<td>-443.22</td>
<td>-571.92</td>
<td>-242.32</td>
<td>-300.74</td>
<td>* Ki</td>
</tr>
<tr>
<td>810.54</td>
<td>131.79</td>
<td>34.43</td>
<td>82.26</td>
<td>* Kd</td>
</tr>
<tr>
<td>170.75</td>
<td>10.41</td>
<td>45.33</td>
<td>15.46</td>
<td>* Kp * Ki</td>
</tr>
<tr>
<td>-587.08</td>
<td>-116.38</td>
<td>-14.25</td>
<td>-40.70</td>
<td>* Kp * Kd</td>
</tr>
<tr>
<td>-48.71</td>
<td>9.49</td>
<td>-3.47</td>
<td>2.55</td>
<td>* Ki * Kd</td>
</tr>
<tr>
<td>810.15</td>
<td>179.35</td>
<td>41.38</td>
<td>89.08</td>
<td>* Kp²</td>
</tr>
<tr>
<td>352.05</td>
<td>452.48</td>
<td>141.18</td>
<td>218.05</td>
<td>* Ki²</td>
</tr>
<tr>
<td>47.00</td>
<td>16.95</td>
<td>0.99</td>
<td>-0.75</td>
<td>* Kd²</td>
</tr>
<tr>
<td>39.11</td>
<td>6.09</td>
<td>0.45</td>
<td>8.24</td>
<td>* Kp * Ki * Kd</td>
</tr>
<tr>
<td>-45.30</td>
<td>-13.19</td>
<td>-4.90</td>
<td>-12.28</td>
<td>* Kp² * Ki</td>
</tr>
<tr>
<td>123.77</td>
<td>26.99</td>
<td>2.65</td>
<td>4.05</td>
<td>* Kp² * Kd</td>
</tr>
<tr>
<td>-14.25</td>
<td>10.53</td>
<td>-10.11</td>
<td>10.39</td>
<td>* Kp * Ki²</td>
</tr>
<tr>
<td>-17.96</td>
<td>-8.17</td>
<td>-0.50</td>
<td>1.77</td>
<td>* Kp * Kd²</td>
</tr>
<tr>
<td>6.21</td>
<td>-1.08</td>
<td>1.71</td>
<td>-2.66</td>
<td>* Ki² * Kd</td>
</tr>
<tr>
<td>10.44</td>
<td>-1.57</td>
<td>-0.42</td>
<td>-1.19</td>
<td>* Ki * Kd²</td>
</tr>
<tr>
<td>-131.37</td>
<td>-29.00</td>
<td>-5.33</td>
<td>-10.78</td>
<td>* Kp³</td>
</tr>
<tr>
<td>-112.28</td>
<td>-129.34</td>
<td>-31.99</td>
<td>-63.40</td>
<td>* Ki³</td>
</tr>
<tr>
<td>2.01</td>
<td>2.18</td>
<td>0.15</td>
<td>0.34</td>
<td>* Kd³</td>
</tr>
</tbody>
</table>
**Fig. 5.** Surface plots of all four metamodels over PID gain parameters (input variables).

**Fig. 6.** Scatterplots over standardized residuals for all four metamodels.
Table 4. Optimal points derived through different tuning methods.

<table>
<thead>
<tr>
<th>Tuning method</th>
<th>Gain parameters</th>
<th>Performance</th>
<th>Rise time</th>
<th>Settling time</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K_p$ $K_i$ $K_d$</td>
<td>ITAE ITSE ISE IAE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ziegler-Nichols</td>
<td>1.59 1.00 0.63</td>
<td>501 97 103 203</td>
<td>3.21</td>
<td>10.70</td>
<td>4.82</td>
</tr>
<tr>
<td>Taguchi-GRA</td>
<td>2.00 0.80 0.60</td>
<td>349 83 96 182</td>
<td>3.90</td>
<td>6.24</td>
<td>0.77</td>
</tr>
<tr>
<td>MBSO (ITAE)</td>
<td>1.99 0.87 0.21</td>
<td>288 69 88 166</td>
<td>3.35</td>
<td>5.10</td>
<td>1.26</td>
</tr>
<tr>
<td>MBSO (ITSE,ISE,IAE)</td>
<td>1.98 1.19 0.20</td>
<td>303 59 80 158</td>
<td>2.59</td>
<td>8.90</td>
<td>4.77</td>
</tr>
</tbody>
</table>

Results have confirmed the superiority of MBSO with combining polynomial regression and LHS method has compared with traditional Ziegler-Nichols method and Taguchi-GRA method. In order to reduce the computational cost, the MBSO method tries to approximate and analyze the model with a minimum number of simulation running that for the current model is equal to 30 simulation runs for training and 20 runs for validating metamodel. In the current example, the optimization model is constructed by minimizing performance criterion as a single-objective of the model (minimization of ITAE, ITSE, ISE, or IAE, separately). Results confirm that for each criterion in the presented optimal point, the single objective of the model is satisfied. The performance parameters, including rise time, settling time, and percent overshoot for all methods are shown in Fig. 7.

Fig. 7. Performance parameters based on MBSO method and two classic methods (Ziegler-Nichols and Taguchi-GRA).
We gain the step responses of DC motor speed with PID gain parameters that respectively have obtained from traditional Ziegler-Nichols method, Taguchi-GRA, MBSO (ITAE), and MBSO (ITSE, ISE, IAE) method; see Fig. 8.

**Fig. 8.** Step responses with four different PID gain parameters obtain with Ziegler-Nichols, Taguchi-GRA, and MBSO.

5. Conclusion

In the current paper, the MBSO algorithm was applied for optimal design of PID controller by using hybrid polynomial regression as a metamodel and LHS method for designing of simulation experiments. Three types of sample points were produced by LHS including training points, validating points, and grid search points. The applicability and superiority of proposed method compared with two traditional Ziegler-Nichols method and Taguchi-GRA method were shown by employing practical case in PID tuning for linear speed control of DC motor. Adapting proposed method by other metamodels such as Kriging and RBF can be considered for future researches.

References


Design and analysis of computer experiments using polynomial regression and latin hypercube...