Original Paper

MEASUREMENT AND CONTROL

Measurement and Control
2019, Vol. 52(7-8) 888–895
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0020294019830108
journals.sagepub.com/home/mac
SAGE

Controller parameter optimization for complex industrial system with uncertainties

Heping Chen[1], Seth Bowels[1], Biao Zhang[2] and Thomas Fuhlbrigge[2]

Abstract
Proportional–integral–derivative control system has been widely used in industrial applications. For complex systems, tuning controller parameters to satisfy the process requirements is very challenging. Different methods have been proposed to solve the problem. However these methods suffer several problems, such as dealing with system complexity, minimizing tuning effort and balancing different performance indices including rise time, settling time, steady-state error and overshoot. In this paper, we develop an automatic controller parameter optimization method based on Gaussian process regression Bayesian optimization algorithm. A non-parametric model is constructed using Gaussian process regression. By combining Gaussian process regression with Bayesian optimization algorithm, potential candidate can be predicted and applied to guide the optimization process. Both experiments and simulation were performed to demonstrate the effectiveness of the proposed method.

Keywords
Proportional–integral–derivative control, controller parameter optimization, Gaussian process regression, Bayesian optimization

Date received: 14 March 2018; accepted: 13 January 2019

Introduction

In industrial applications, Proportional–integral–derivative (PID) control is deployed in over 90% systems because it is very easy to use and relatively robust. However, tuning the PID parameters to achieve required process requirements is not an easy task due to the following reasons: (1) several performance indices, including overshoot, rise time, steady-state error and settling time must be evaluated at the same time; (2) the relationship between the PID parameters and these performance indices is unknown; (3) it is not realistic to perform many experiments to tune controller parameters due to cost and safety-related issues. Therefore, developing an efficient method to tune PID controller parameters is worth investigating.

There are some rule-based methods developed to tune PID controller parameters, such as Tyreus–Luyben method, Cohen and Coon method, Ziegler–Nichols method, Ciancone–Marline method, internal model control (IMC) method and C-H-R method.[1] Some disadvantages of these methods are as follows: (1) obtaining optimal PID controller parameters is challenging; (2) it is difficult to consider all performance indices simultaneously; (3) controller parameter tuning process could be affected by disturbances and noise. For complex dynamic systems, it is even harder to achieve optimal system performance using rule-based methods.

To overcome the problems of rule-based methods, some optimal parameter optimization methods are explored. Based on a soft computing optimization method, Matousek et al.[2] developed an optimal PID controller parameter tuning method. Compared to some existing methods, the proposed method is quite effective. Particle swarm optimization method and genetic algorithms are also proposed[3] to tune controller parameters. However, these methods are implemented offline and local optimal parameters could be identified.

Some machine-learning methods are also investigated to automatically tune the controller parameters.

[1]Ingram School of Engineering, Texas State University, San Marcos, TX, USA
[2]ABB Inc., Windsor, CT, USA

Corresponding author:
Heping Chen, Ingram School of Engineering, Texas State University, San Marcos, TX 78746, USA.
Email: hc15@txstate.edu

For some critical systems, automatic parameter tuning may cause safety issues;[4] hence, a safety critical controller parameter tuning method is investigated using safe Bayesian optimization method. Because neural networks are very powerful to construct a model, they are used to explore optimal controller parameters.[5,6] However, many data sets must be collected to train the networks. For industrial applications, it is not a good idea to perform many experiments considering the manufacturing cost, labor, safety and so on. Chan et al.[7] proposed a Gaussian process to predict the optimal performance of a system. It is a very interesting method; however, the balancing method between modeling and prediction is not discussed; furthermore, the performance indices are not evaluated simultaneously.

Since the PID controller parameters must be tuned on the real system, it would be ideal if the controller parameters could be obtained by performing a few experiments. Hence, it is worth investigating a more efficient parameter optimization method. However, tuning the PID controller parameters online is very challenging due to the following reasons:

- Industrial processes are typically very complex. It is very difficult to obtain optimal controller parameters;
- Constructing a model using a few data sets to predict optimal parameters is a good way to solve the problem, but local minima could be generated if the data sets cannot represent the system;
- There are always uncertainties and noise in the data collection;
- Multiple performance indices must be considered simultaneously in parameter tuning process.

Considering the challenges and requirements in PID controller parameter tuning, we propose a new method based on Gaussian Process Regression (GPR) and Bayesian optimization algorithm (BOA). By combining these two methods, a system model can be constructed iteratively using GPR, and potential candidate can be predicted at each iteration using BOA. Hence, the combined GPRBOA method can guide the optimization process and thus reduce the number of experiments. To avoid local convergence, a method to investigate unexplored region is developed. In order to validate the proposed method, we performed simulation as well as experiments. Multiple performance indices including settling time, steady-state error, rise time and overshoot are evaluated during the optimization process. A multi-objective optimization problem is formulated to consider all performance indices simultaneously. Uncertainties and noise were added to the simulation and experiments. The results demonstrate that the proposed method is very effective in PID controller parameter optimization.

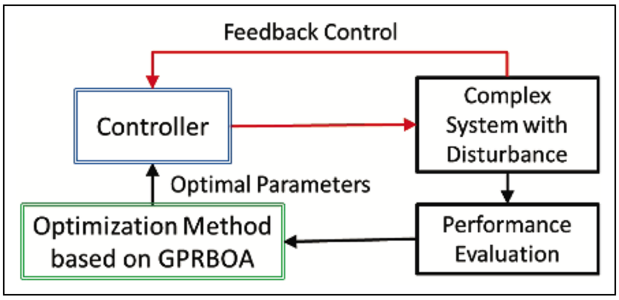

Figure 1. Framework of proposed controller parameter optimization method.

Proposed solution

The proposed framework of optimizing PID controller parameters is illustrated in Figure 1.

Given a set of PID controller parameters, the output of a complex system can be obtained. The performance indices such as rise time, overshoot, steady-state errors and settling time can then be evaluated. Based on the performance evaluation, GPRBOA can generate a potential optimal candidate to perform the experiment again. When the system response satisfies the desired system performance requirements, the process stops. Hence, there are two major problems we have to deal with: evaluation of performance indices and controller parameter optimization method based on GPRBOA.

Performance indices

When considering the controller parameters as the input to a complex system, the structure of the system with multiple input and output can be illustrated in Figure 2.

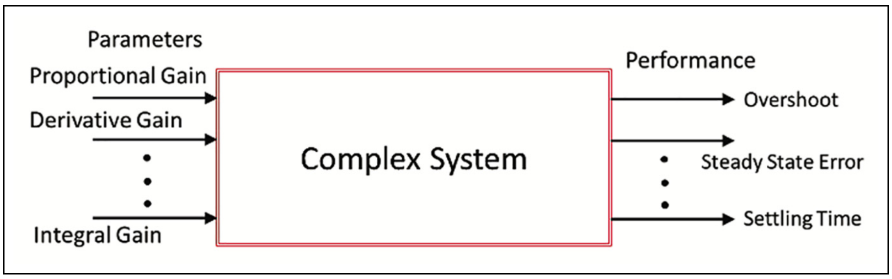

Figure 2. The structure of a system with multiple input and multiple output.

For a system with several performance indices, they must be balanced. Considering safety-related issues, we may need to stop experiment when some performance indices are out of range or the system becomes unstable. Hence a multi-objective optimization problem with constraints is formulated

$$\begin{aligned} &F(x) = \{F_1(x), F_2(x), \ldots, F_n(x)\} \\ &\text{Subject to}: F_i^{\min} < F_i < F_i^{\max} \end{aligned} \tag{1}$$

where F_i is the ith performance index ($i = 1, \ldots, n$); $F_i^{\min}$ and $F_i^{\max}$ are the minimum and maximum values of the performance indices, respectively. Different methods can be applied to solve the constraint multi-objective optimization problem. In this paper, the following method is adopted to convert the constraint multi-objective optimization problem into a single objective optimization problem

$$\begin{aligned} y(x) = &\sum_{j=1}^{n} w_j F_j(x) - \sum_{i=1}^{n} U_i \min\left(0, F_i(x) - F_i^{\min}\right) \\ &- \sum_{i=1}^{n} V_i \min\left(0, F_i^{\max} - F_i(x)\right) \end{aligned} \tag{2}$$

where $y(x)$ is the output; w_j is the weight; U_i and V_i are large random numbers used to deal with the issue that the performance indices are out of range. w_j can be assigned such that some performance indices are emphasized.

GPR

GPR is an effective tool to construct models for complex systems.[8,9] It is a non-parametric method which can deal with uncertainties and noisy observations.[10]

For a complex process with PID control, we assume that the relationship between the output (performance indices) and the input (PID controller parameters) can be written as

$$y = f(x) + \epsilon \text{ with } \epsilon \sim N\left(0, \sigma^2\right) \tag{3}$$

$f(x)$ is a Gaussian process which can be defined as

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{4}$$

where $\mathcal{GP}$ represents Gaussian process; $m(x)$ is the mean function; $k(x, x')$ is the covariance function; and x and x' are two sets of controller parameters. The covariance function is defined as

$$k(x, x') = E[(m(x) - f(x))(m(x') - f(x'))] \tag{5}$$

After performing a series of experiments, the output of a system with PID control can be evaluated using equation 2. We can then obtain a set of data (X, y) with

$$\begin{aligned} X &= [x_1, x_2, \ldots, x_m] \\ y &= [y_1(x_1), y_2(x_2), \ldots, y_m(x_m)] \end{aligned} \tag{6}$$

The covariance function is very important in constructing a model and predicting system output. Given a data set (X, y) and a data set to be predicted (X^*, y^*), their covariance function can be described as

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \begin{bmatrix} (X, X^*) + \sigma^2 I & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix} \tag{7}$$

where $K(X', X'')$ is the covariance matrix whose element $k_{ij} = k(x_i', x_j'')$.

By deriving the conditional distribution, we can predict the mean and variance of the system output

$$y^* | X, y, X^* \sim N(\mu(y^*), \mathrm{V}(y^*)) \tag{8}$$

$$\begin{aligned} \mu(y^*) &= K(X^*, X)\Lambda^{-1} y \\ \mathrm{V}(y^*) &= K(X^*, X^*) - K(X^*, X)\Lambda^{-1} K(X, X^*) \end{aligned} \tag{9}$$

with

$$\Lambda = K(X, X) + \sigma^2 I \tag{10}$$

where $\mu(y^*)$ is the predicted mean and $\mathrm{V}(y^*)$ is the predicted variance for y^*.

Equation (9) tells that the covariance function $k(x, x')$ is the key to predict the output of a complex system with PID control. Different covariance functions can be used for GPR. In this paper, we combine different covariance functions: constant valued covariance function (Cov_1), squared exponential covariance function with isotropic distance measure (Cov_2) and white noise covariance function (Cov_3).[10] The combination method is

$$k(x, x') = \lambda_1 Cov_1 + \lambda_2 Cov_2 + \lambda_3 Cov_3 \tag{11}$$

where $\lambda_i (i = 1, 2, 3)$ are arbitrary constants. In the covariance function, there are hyperparameters θ. In order to obtain the best fit to the observed data set (X, y), θ must be optimized. Considering the hyperparameters and given data set, the posterior probability of the function $f(x)$ can be computed using

$$p(f | X, y, \theta) = \frac{p(y | X, f, \theta) p(f | X, \theta)}{p(y | X, \theta)} \tag{12}$$

The marginal likelihood can be calculated using

$$p(y | X, \theta) = \int p(y | X, f, \theta) p(f | X, \theta) df \tag{13}$$

Because the system is assumed to be a Gaussian process, the log marginal likelihood can be solved analytically. The following equation is then obtained

$$\log p(y | X, \theta) = -\frac{1}{2} y^{\mathrm{T}} (\Lambda)^{-1} y - \frac{1}{2} \log|\Lambda| - \frac{n}{2} \log(2\pi) \tag{14}$$

By maximizing the marginal log likelihood, we can obtain the optimal hyperparameters

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log p(y | X, \theta) \tag{15}$$

Once the hyperparameters are determined, the covariance function $k(x, x', \theta)$ is known. For a set of PID parameters x^*, we can then predict y^* using equation 9. The mean and variance of y^* can be obtained.

Performance evaluation

Once the mean and variance of y^* are obtained for a given set of input x^*, the problem becomes how to explore an optimal set of PID controller parameters based on the mean and variance. An acquisition function is adopted to evaluate the system performance. There are different methods to formulate an acquisition function such as probability of improvement,[11] expected improvement[12] and upper confidence bound (UCB).[13] In this paper, we use UCB to evaluate the system performance based on the mean and variance. The UCB is defined as

$$UCB(x)\Delta = \mu(x) + \zeta \cdot \sigma(x) \tag{16}$$

where ζ is a scaling factor. When UCB is minimized, the output of the system is minimized. A set of optimal PID controller parameters is identified.

Exploration and exploitation

When building a model using Gaussian process, a few data sets are required at the beginning to make sure the model is reasonable; meanwhile, local convergence must be avoided during the optimization process. In order to solve the problem, we must generate some data sets based on exploration. To minimize the number of experiments and expedite the optimization process, we must generate candidate set of controller parameters based on predicted optimal controller parameters. Therefore, these two processes must be balanced. GPRBOA can switch between the exploration process and exploitation process to update the system model and perform optimization. For the exploitation process using equation 16, the prior information is required, but it is unknown when the system model is not available. Hence, a balancing method is proposed in this paper to control the exploration process and exploitation process to generate candidate set of controller parameters

$$x^* = \begin{cases} \arg\min UCB(x) & \text{rand}(1) > \beta \\ \arg\max \psi(x) & \text{otherwise} \end{cases} \tag{17}$$

where β is updated during the optimization process using the hyperparameters

$$\beta = \begin{cases} 1 & \Delta\theta > \theta_{th} \\ \Delta\theta & \text{otherwise} \end{cases} \tag{18}$$

where θth is a threshold; $\Delta\theta$ is computed using the following equation

$$\Delta\theta = \frac{|\theta_k - \theta_{k-1}|}{|\theta_{\max} - \theta_{\min}|} \tag{19}$$

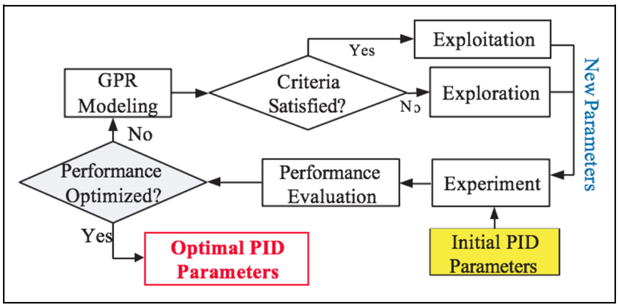

Figure 3. Implementation of GPRBOA for PID controller parameter optimization.

where θ_k is the hyperparameters at the kth iteration; $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum values in the optimization process. When $\Delta\theta > \theta$th, the unsampled parameter region will be explored. $\psi(x)$ is applied to find a candidate by exploring the unsampled region

$$\psi(x) = \min_{i \in 1, \dots, N} d(x - x_i) \tag{20}$$

where $d(x - x_i)$ is defined using two PID parameter sets x and x_i. At each iteration, if rand(1) $< \beta$, the candidate is exploited based on equation 16 by minimizing UCB; otherwise, it is generated by exploring the farthest unsampled region.

GPRBOA implementation

GPRBOA balances the exploration process and exploitation process. At each iteration, new data sets are added into the existing data sets to update the system model. The updated model is then used to predict the mean and variance of the system output. The process of implementing GPRBOA is shown in Figure 3.

After performing experiments using the initial set of parameters, the system output is evaluated using equation 16. If the system performance does not satisfy the desired system requirements, the system model is updated using the data sets. The hyperparameters are then evaluated to balance the exploration and exploitation processes. A new candidate is then generated. Using the new set of controller parameters, the experiment is performed again. The process stops once an optimal set of controller parameters is found.

Simulation results

In order to validate the proposed method, we performed simulation first. Even though a linear system was used for simulation, we added random noise to each model parameter. The added noise makes the system nonlinear. The simulation was implemented using MATLAB. The implemented transfer function is

$$G(s) = \frac{a_2 s^2 + a_1 s + a_0}{b_5 s^5 + b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \tag{21}$$

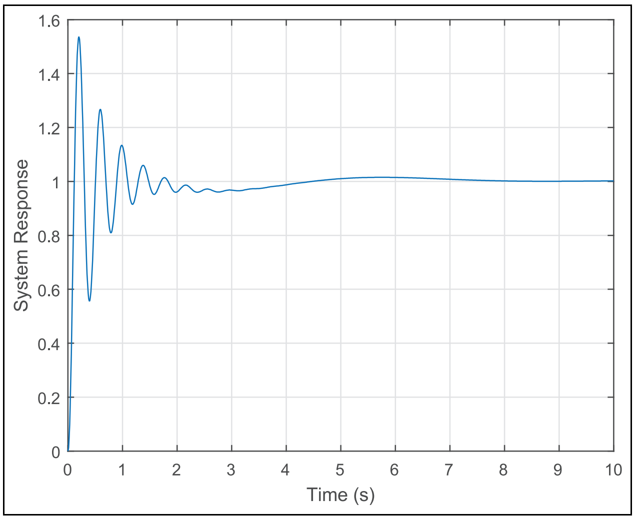

Figure 4. The step response for a system with noise.

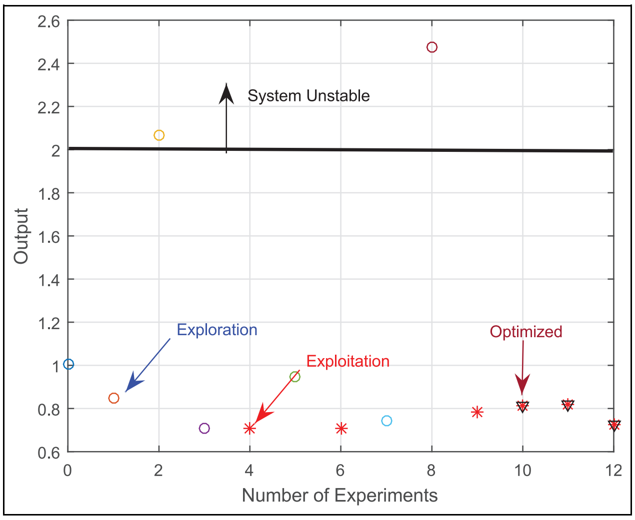

Figure 5. The controller parameter optimization process for a system with noise. The optimal set of parameters were identified using 10 experiments.

with

$$a_0 = 3; a_1 = 15; a_2 = 10;$$
$$b_0 = 3; b_1 = 25; b_2 = 1; b_3 = 20; b_4 = 3; b_5 = 0.5. \quad (22)$$

In the simulation, a white noise (signal-to-noise ratio 30 dB) is added to the model parameters (a_i and b_i). A 30 dB noise is also added to the system output to simulation the output noise and uncertainties.

The ranges of the PID parameters are set as

$$K_p \in [1020]; \; K_i \in [515]; \; K_d \in [120] \quad (23)$$

The step size is 0.5. We used the following weights in order to minimize the steady-state error

$$w_1 = 0.01; \; w_2 = 0.05; \; w_3 = 0.1; \; w_4 = 0.84 \quad (24)$$

The step response of the system with noise is shown in Figure 4. The optimal set of PID parameters are

$$K_p = 10.5; \; K_i = 9.5; \; K_d = 11.5 \quad (25)$$

The performance indices are as follows: overshoot 53.5%; settling time 3.7 s; rise time 0.08 s and steady-state error in the simulation range is −0.0018.

The optimization process is shown in Figure 5. There are exploration and exploitation processes in the experiments. The optimal set of controller parameters was found using 10 simulations. The simulation continued using the identified optimal controller parameters with random noise added to the model parameters. If the system was not stable, a random number greater than two was added to the output.

Different transfer functions with added noise were also simulated. Even though the number of experiments exploring optimal set of parameters may be different, the performance of the proposed method is quite consistent.

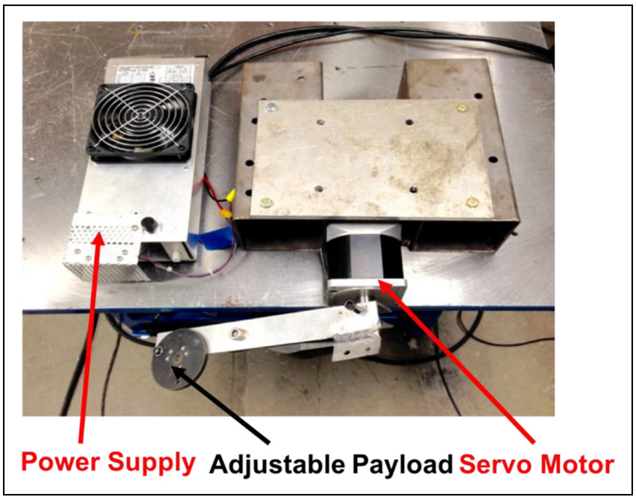

Figure 6. The motor control system used to demonstrate the proposed method.

Experimental results

Experiments were also performed to evaluate the proposed method. A servo motor position control system was developed to demonstrate the controller parameter tuning process. The experimental system is shown in Figure 6. The servo motor (Smart Motor SM34405D) with a 4:1 high precision gear box is mounted on a work table. An arm with a fixed weight is installed on the motor shaft. The proposed method is implemented on a computer, which receives the system response from the controller and generates candidates for the controller. To demonstrate the robustness of the proposed method, an adjustable weight to change the motor payload is added. The location of the adjustable weight can be changed to simulate the disturbance to the controller. The ranges of the PID controller parameters are set in the computer program.

Table 1. Ranges of performance indices.

F_1 (s)	F_2(s)	F_3 (°)	F_4 (°)
[0, 1]	[0, 5]	[0, 5]	[0, 0.5]

Table 2. PID controller parameters and four performance indices.

K_p	K_i	K_d	F_1	F_2	F_3	F_4	$y(\mathbf{x})$
1	0.1	2	0.200	0.383	0.640	0.000	0.186
10	1	50	0.204	0.291	0.120	0.053	0.105
10	0.1	1	0.212	0.498	0.070	0.032	0.104
1	1	26	0.285	1.000	7.010	0.530	1.849
9	0.9	2	0.185	0.520	0.760	0.201	0.343
2	0.2	50	0.206	0.303	0.160	0.042	0.108
5	0.5	20	0.206	0.303	0.220	0.000	0.095
5	0.5	20	0.212	0.244	0.260	0.074	0.142
7	0.5	50	0.219	0.450	0.330	0.170	0.235
5	0.5	20	0.204	0.300	0.230	0.000	0.096
7	0.1	2	0.423	0.550	0.410	0.053	0.211
5	0.5	20	0.206	0.296	0.230	0.000	0.096
5	0.5	20	0.207	0.301	0.220	0.000	0.095
5	0.5	20	0.209	0.300	0.220	0.000	0.095
5	0.5	20	0.206	0.299	0.220	0.000	0.095

Note: The values of K_p, K_i and K_d in Table 2 should multiply by 1000. F_1 is the rise time (s); F_2 settling time (s); F_3 overshoot (°) and F_4 steady-state error (°).

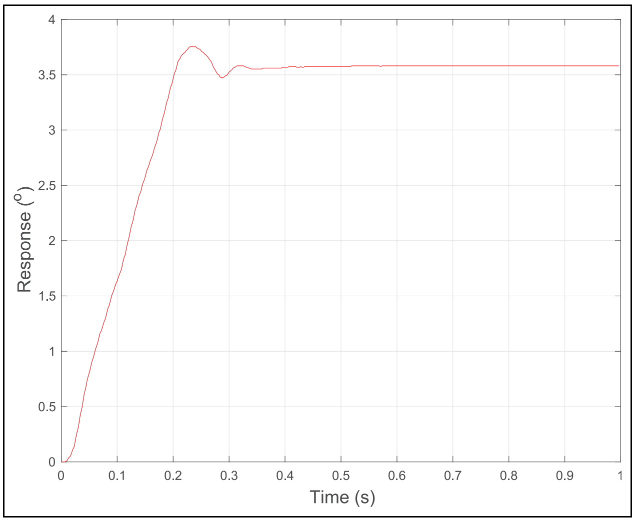

Figure 7. The step response using optimal parameters obtained using GPRBOA for a system with noise.

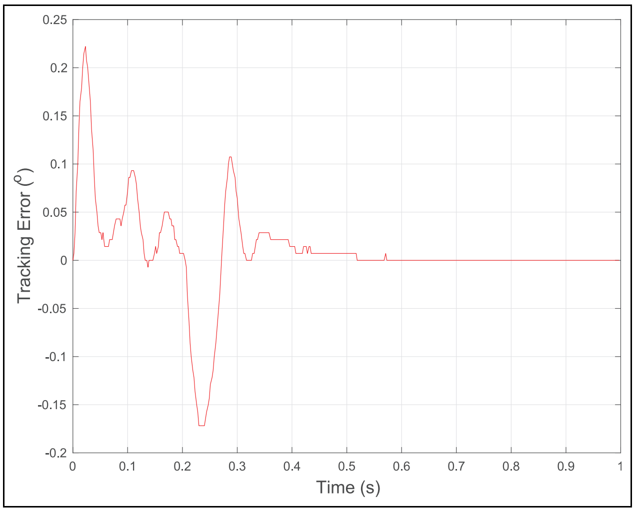

Figure 8. The position-tracking error using optimal parameters.

At each experiment, the motor moves about 3.6° using a fixed velocity and acceleration. The maximum velocity is set to be 3.75 rev/s while maximum acceleration 2.5rev/s^2. We recorded the response of the system. The rise time (F_1), settling time (F_2), overshoot (F_3) and steady-state error (F_4) were obtained from the recorded data. Based on these performance indices, the performance index of the system is evaluated using equation 2. U_i and V_i are set to be 100. Table 1 shows the minimum and maximum values of each performance index.

Because the steady-state error is more important than the other performance indices, we used the following weights

$$w_1 = 0.1;\ w_2 = 0.1;\ w_3 = 0.2;\ w_4 = 0.6 \qquad (26)$$

To generate new candidate, the range of each PID controller parameter must be set

$$K_p \in [100010000];\ K_i \in [1001000];\ K_d \in [200050000]$$

The step size is important for computational time. The step sizes of the three PID controller parameters are 1000, 100 and 2000 for proportional gain, integral gain and derivative gain, respectively.

After an experiment is performed, its performance index is evaluated using equation 2. The result is then sent to the computer. The proposed method will generate a new candidate set of controller parameters and the candidate set is sent to the robot controller. The experiment will be performed again. This process continues until a set of optimal controller parameters is identified.

Table 2 shows the experimental results. For some cases, such as $K_p = 1000;\ K_i = 1000$ and $K_d = 26,000$, the system is not stable and not settled down. A large random number is then added to the output.

The system response using optimal set of controller parameters is shown in Figure 7. The optimal set of PID controller parameters is

$$K_p = 5000;\ K_i = 500 \text{ and } K_d = 20,000 \qquad (27)$$

The position tracking error of the optimal set of controller parameters is shown in Figure 8. The rise time is 0.21 s, settling time 0.3 s, overshoot 0.22° and steady state error 0. The performance index is 0.095.

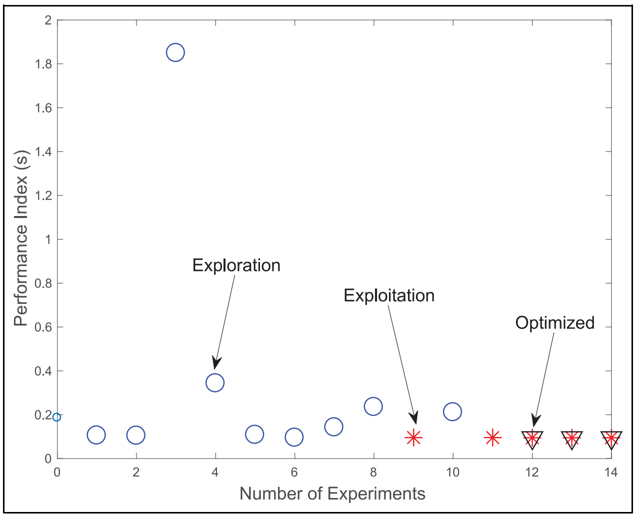

Figure 9. The experimental results for PID controller parameter optimization process. An optimal set of controlled parameters was identified using 11 experiments.

Figure 9 shows the performance indices and the number of experiments performed to find the optimal PID parameters (11 experiments). We can see that the exploration and exploitation processes are involved in parameter optimization process. Once the optimal set of PID parameters is found, the experiment continues to verify the performance of the optimal set of controller parameters for the system with random payload.

We performed many experiments using different payloads and the results are quite consistent. Therefore, the proposed method is an effective tool to explore an optimal set of controller parameters to satisfy system performance requirements.

Compared to other existing methods, our proposed method has several advantages:

- Compared to the method developed by Chan et al.,[7] our method considers several performance indices simultaneously and develops a balancing method to avoid local convergence.
- Compared to the methods based on neural network,[5,6] our method guides the optimization process, thus requires fewer experiments.
- Compared to the rule-based methods,[1] our method does not require human involvement and can achieve optimal solution.

Conclusion

In this paper, an online PID controller parameter optimization method is developed based on GPRBOA. The system model and potential optimal candidate are updated iteratively. The exploration and exploitation processes are balanced using a proposed acquisition function method. A multi-objective optimization method is formulated. To validate the proposed methods, both simulation and experiment were performed. The results demonstrate the proposed method can be used to guide the PID controller parameter tuning process to achieve optimal output. The proposed method is based on non-parametric modeling and optimization algorithms, thus it can be applied to optimize process parameters for other complex systems.

Acknowledgements

The authors would like to thank the Ingram School of Engineering, Texas State University, for supporting the research.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The project is partially supported by the Program of Foshan Innovation Team of Science and Technology (grant no. 2015IT100072). The project is partially supported by ABB Inc.

ORCID iD

Heping Chen https://orcid.org/0000-0002-1045-8977

References

1. Hussain KM, Zepherin RAR and Kumar MS. Comparison of tuning methods of PID controllers for FOPTD system. *Int J Innovat Res Elect Electr Instrument Contr Eng* 2014; 2: 1177–1180.
2. Matousek R, Minar P, Lang S, et al. Hc12: Efficient method in optimal PID tuning. In: *Proceedings of the world congress on engineering and computer science*, San Francisco, CA, 19–21 October 2011.
3. Jaen-Cuellar AY, Romero-Troncoso RJ, Morales-Velazquez L, et al. PID-controller tuning optimization with genetic algorithms in servo systems. *Int J Innovat Res Elect Electr Instrument Contr Eng* 2013; 10: 1–14.
4. Berkenkamp F, Schoellig AP and Krause A. Safe controller optimization for quadrotors with Gaussian processes. In: *IEEE international conference on robotics and automation*, vol. 3, Stockholm, 16–21 May 2016, pp. 491–496. New York: IEEE.
5. Reichensdörfer E, Günther J and Diepold K. Recurrent neural networks for PID auto-tuning: adaptive control and identification of nonlinear systems, 2017. *Technical Report*. http://mediatum.ub.tum.de/doc/1381851/534530033346.pdf
6. Zribi A, Chtourou M and Djemel M. A new PID neural network controller design for nonlinear processes. *J Circuits Syst Comp* 2017; 27(4): 11.
7. Chan L, Chen T and Chen J. Applying Gaussian process models to on-line enhancement of PID control design for nonlinear processes. In: *5th international symposium on*

advanced control of industrial processes, Hiroshima, Japan, 28–30 May 2014, pp. 114–119.

8. Yang K, Keat Gan S and Sukkarieh S A Gaussian process based RRT planner for the exploration of an unknown and cluttered environment with a UAV. *Adv Robot* 2013; 27(6): 431–443.
9. Jadaliha M, Xu Y, Choi J, et al. Gaussian process regression for sensor networks under localization uncertainty. *IEEE Trans Signal Proc* 2013; 61(2): 223–237.
10. Rasmussen CE and Williams CKI. *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press, 2006.
11. Hawe G and Sykulski J. An enhanced probability of improvement utility function for locating pareto optimal solutions In: *16th conference on the computation of electromagnetic fields (COMPUMAG)*, vol. 3, Aachen, 24–28 June 2007, pp. 965–966. Aachen: RWTH Aachen University.
12. Kleijnen J, Beers W and Nieuwenhuyse I. Expected improvement in efficient global optimization through bootstrapped Kriging. *J Global Optimizat* 2012; 54(1): 59–73.
13. Garivier A and Moulines E. On upper-confidence bound policies for switching bandit problems. In: *Proceedings of the 22nd international conference on algorithmic learning theory (ALT'11)*, Espoo, 5–7 October 2011, pp. 174–188. Berlin; Heidelberg: Springer.