

PAPER • OPEN ACCESS

Optimization of RTD-A Controller Parameters Based on Empirical Bat Algorithm

To cite this article: Kebin Lu *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **569** 042023

View the [article online](#) for updates and enhancements.

Optimization of RTD-A Controller Parameters Based on Empirical Bat Algorithm

Kebing Lu¹, Jizheng Chu^{1*} and Minghui Chu¹

¹ College of Information Science & Technology, Beijing University of Chemical Technology, Beijing, 100000, China

*Corresponding author's e-mail: chujz@mail.buct.edu.cn

Abstract. During recent twenty years, the research of predictive control has become mature and has strong application value. It was over this period that the RTD-A controller was proposed as a new predictive controller with a simple structure and easy implementation. The four parameters of the controller directly affect the robustness of the system, the traceability of the set points, the suppression of interference and the global convergence, making the control effect of RTD-A better than that of the classic controller. However, it is found in the controller parameter modulation of the industrial field that although the four parameters are clear, the coupling phenomenon between the parameters is reflected in the system of set point tracking and interference suppression, which increases the difficulty of parameter modulation. Based on this, this paper proposes offline intelligent parameter optimization for the process of setpoint tracking and interference suppression in a finite step range. Finally, it is proved by simulation experiments that the RTD-A controller using the intelligent optimization algorithm has better setpoint tracking performance, stronger anti-interference ability and better robustness than the RTD-A controller without the optimization algorithm.

1. Introduction

Since the Model Predictive Control(MPC) appeared in the field of industrial process control in the 1970s, predictive control has been fully developed. During the period, Ogunnaike proposed a new predictive controller RTD-A(Robustness, Tracking, Disturbance rejection - overall Aggressiveness)[1]. The controller model has a simple structure; as far as the principle principle is implemented, the controller has the same characteristics as the classical predictive controller: model prediction, rolling optimization and feedback correction; The most important thing is that the controller parameters directly affect the relevant performance of the controller. These make the RTD-A controller have a prominent advantage in the field of robust control and parameter tuning.

While the predictive control is booming, it also stimulates the enthusiasm for optimizing predictive control research. A large number of excellent personnel have made outstanding contributions to the optimization of predictive control. For example, Danielle proposed a dynamic matrix control (MPC)tuning guide for the lumping process and gave a specific explicit formula[2]. McIntosh et al. proposed the range of tuning parameters based on simulations of process lag and rise time[3, 4]. These studies have made no small contribution to the development of optimal predictive control. Research has not stopped. In recent years, a large number of researchers have used intelligent optimization algorithms or recursive algorithms to optimize and modulate offline or online parameters. For example, Bououden et al. proposed a scheme of ant colony optimization for the fuzzy predictive control model of nonlinear processes[5]. Liu Wei et al. proposed the idea of automatic tuning of recursive formulas



based on model predictive control[6]. These ideas also provide a reference direction for the modulation of the new predictive controller RTD-A parameters.

2. Mathematical Preliminaries

2.1. Principle of RTD-A controller

The first-order plus delay model (FOPDT) is easy to identify and has a wide representation. Therefore, the internal model of many classical controllers generally adopts the first-order plus delay model (FOPDT) model. Referring to the classic, the internal model of the predictive controller RTD-A is also expressed using the first order plus delay model (FOPDT).

$$\hat{y}(s) = \frac{K e^{-\alpha s}}{\tau s + 1} u(s) \quad (1)$$

Where: K is the steady state gain; τ is the effective time constant; α is the effective dead time.

The first-order plus delay model (FOPDT) discretization is expressed as:

$$\hat{y}(k+1) = a \hat{y}(k) + b u(k-m) \quad (2)$$

In the formula, $k = 0, 1, 2, \dots$; $a = e^{-\Delta t/\tau}$; $b = K(1-a)$; $m = \text{round}(\alpha/\Delta t)$, is the delay time; Δt is the sampling time; $\hat{y}$ is the output value of the model. Through the recursive operation, the following expressions of the future N time step prediction outputs can be obtained:

$$\hat{y}(k+m+i) = a^{m+i} \hat{y}(k) + a^{i-1} b \mu(k, m) + b \eta_i u(k) \quad 1 \leq i \leq N \quad (3)$$

In the formula, $\mu(k, m) = \sum_{i=1}^m a^i u(k-i)$; $u(k+i) = u(k)$, $k = 1, 2, 3, \dots, N$; $\eta_i = \frac{1-a^j}{1-a}$.

$e(k) = y(k) - \hat{y}(k)$, the error $e(k)$ is decomposed into two parts, One part represents the non-biased residual of system integrity; the other part represents the deviation of the model from the real object. In the case where the deviation prior distribution and sampling distribution are known, the maximum posterior distribution estimate (MAP) of the deviation is obtained according to Bayes' theorem.

$$\hat{D}(K) = \theta_r \hat{D}(k-1) + (1-\theta_r) e(k) \quad (4)$$

In the formula, $\theta_r = \sigma_r^2 / (\sigma_r^2 + \nu^2) \in [0, 1]$, σ_r^2 is the variance of the unbiased residual, ν^2 is the variance of the deviation, θ_r is the controller robustness tuning parameter. When θ_r approaches 1, the model develops in the direction of serious mismatch; When θ_r approaches 0, the integrity of the model develops in the exact direction.

The error prediction output expression for the future N time steps is:

$$\hat{D}(k+j|k) = \hat{D}(K) + \frac{1-\theta_d}{\theta_d} [1 - (1-\theta_d)^j] \nabla \hat{D}(k) \quad m+1 \leq j \leq m+N \quad (5)$$

Where $\theta_d = 1-a \in [0, 1]$ is the controller tuning parameter that affects the rapidity of disturbance rejection.

$$y^*(k+j) = \theta_t^j y^*(k) + (1-\theta_t^j) y_d(k) \quad 1 \leq j \leq \infty \quad (6)$$

Where $y_d(k)$ is the ideal set point for the process output $y(k)$; y^* is the set point trajectory that

the controller is using; $\theta_T \in [0,1]$ is the controller tuning parameter affecting the set point tracking speed.

The prediction step N is calculated by the following formula:

$$N = 1 - \frac{\tau}{\Delta t} \ln(1 - \theta_A) \quad (7)$$

According to the above calculation of the process output $\tilde{y}(k+m+j)$ of the future N steps and the desired trajectory $y^*(k+j)$ of the set point, build the following objective function:

$$J(u(k)) = \min_{u(k)} \sum_{i=1}^N (y^*(k+i) - \tilde{y}(k+m+i))^2 \quad (8)$$

According to the principle of the least squares method, the minimum value of the function (8) is solved, that is, the obtained control amount $u(k)$:

$$u(k) = \frac{1}{b} \frac{\sum_{i=1}^N \eta_i \psi_i(k)}{\sum_{i=1}^N \eta_i^2} \quad (9)$$

In the formula: $\psi_i(k) = y^*(k+i) - a^{m+i} y(k) - a^{i-1} b \mu(k, m) - D(k+m+i | k)$.

2.2. Selection of optimization performance function

Predictive controls such as MPC, GPC, DMC, and the predictive controller RTD-A mentioned in this paper all adopt model prediction to obtain the optimal control quantity by solving the sum of squared errors of the future N steps. In this paper [12], the sum of squared error of the next N steps is used as the performance function of parameter optimization. At each step of the control, the intelligent optimization algorithm is used to optimize the parameters and find the optimal controller parameters at the current moment. This is a serious waste of calculation time, and can not meet the rapidity of control performance. When the system is in steady state, the performance function may have multiple solutions, and different control quantities are directly generated, which directly leads to instability of the whole system and is prone to oscillation. Therefore, this paper adopts a method of the offline parameter tuning using the empirical value bat algorithm in a control system with given point variation and external disturbance in a limited time range.

A well-performing control system that reaches or approaches the setpoint quickly, smoothly, and accurately after being subjected to external disturbances or changes in a given value. In order to achieve such a good effect, some unified metrics are needed to evaluate or design the control system. Unit step response curves and error performance indicators are two of the most important metrics. The unit step response curve has six performance indicators: rise time t_r , peak time t_p , overshoot $\sigma\%$, transition process time t_s , attenuation ratio n and steady state error $e(\infty)$. The first five are used to measure the dynamic performance of the control system, The latter is used to measure steady state performance. There are contradictions between some of these performance indicators, such as overshoot and transition process time, so it is impossible to achieve all dynamic performance and steady state performance at the same time. Another measure is the error performance index, which is very suitable for computer simulation and analysis. This paper uses error performance indicators.

In order to obtain the dynamic characteristics of the satisfactory transition process, the error absolute time integral (IAE) performance index is used as the performance function of the controller parameter tuning. In order to prevent the control amount from being too large, the squared term of the control amount is added to the performance function. Use the following formula as a performance

function for parameter optimization:

$$J = \int_0^{\infty} [w_1|e(t)| + w_2u^2(t)]dt + w_3t_r \quad (10)$$

In the formula, $e(t)$ is the difference between the set value and the system output value; $u(t)$ is the control amount of the controller; t_r is the time when the step response curve reaches the new steady state for the first time, that is, the rise time; w_1 , w_2 and w_3 are weight.

In order to avoid overshoot, the penalty function is used. Once overshoot is generated, the overshoot is used as a function of the performance function. so the performance function becomes the following form:

$$J = \int_0^{\infty} [w_1|e(t)| + w_2u^2(t) + w_4|de(t)|]dt + w_3t_r \quad (11)$$

In the formula, $de(t)$ is the difference between the current time and the previous time, and negative in the formula; w_4 is a weight and $w_4 \gg w_1$.

2.3. the principle of the BA bat algorithm

The bat algorithm is a heuristic global search algorithm proposed by Professor Yang in 2010 [1], which combines the main features of the existing particle swarm optimization, simulated annealing and other global search algorithms and the advantages of bat echolocation. In Yang's article, eight examples are used to compare the nonlinear engineering optimization problems with PSO, GA and BA algorithms. The results show that the BA algorithm is superior to PSO and GA algorithms in accuracy and effectiveness.

The bat algorithm combines the bat foraging, obstacle avoidance and positioning functions for global optimization. The specific algorithm process is that each bat individual in the bat population performs flight frequency update, speed update and position update every time a position is reached, and the fitness value of the location is calculated by the fitness function. The bat judges whether or not the prey is found according to the fitness value. Once the prey is found, the loudness decreases, the pulse emission frequency rises, and the position and fitness value are recorded. If the bat enters the local search, it randomly walks around the optimal solution to create a new position. Eventually each bat converges to the same location.

3. simulation

In order to verify the effectiveness of the bat algorithm to optimize RTD-A controller parameters, a simulation experiment was done. The controlled object model used in the simulation is a first-order plus lag model:

$$g(x) = \frac{1 \times e^{-s}}{2s + 1} \quad (12)$$

The parameters of the bat algorithm and the RTD-A controller algorithm are set as follows: the population sample 30 of the bat algorithm; the number of iterations 300 times; the loudness is 0.5; the frequency is 1.0; The sampling time is 0.01s; the RTD-A controller parameter has a value range(0,1); The weight of the performance function is $w_1=0.999, w_2=0.001, w_3=2.0, w_4=100$ (weight can be adjusted according to the optimized effect).

When using the bat algorithm, the parameters are randomly selected within the range of values, which is easy to cause blindness of initial optimization and waste of memory calculation. In this paper, the empirical value of industrial field modulation is used as the initial value of parameter optimization. That is to say, the empirical value is assigned to one of the initial populations of the bat algorithm, which facilitates the iterative convergence of the bat algorithm on the basis of the empirical value. The following is a simulation experiment with empirical value bat algorithm parameters optimization and no parameter optimization. The experimental results are shown in the following figure:

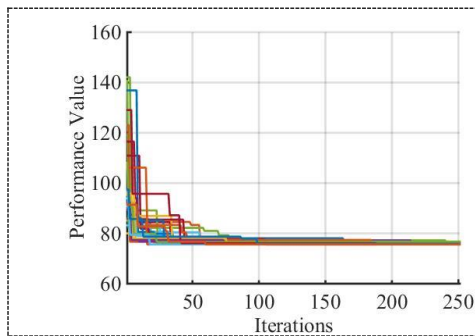


Figure 1. Iterative convergence graph of individuals in bat populations.

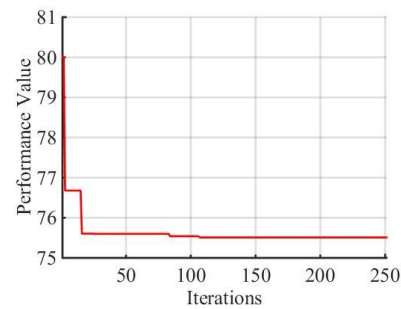


Figure 2. Iterative Convergence Graph of Global Optimal Value in Bat Population.

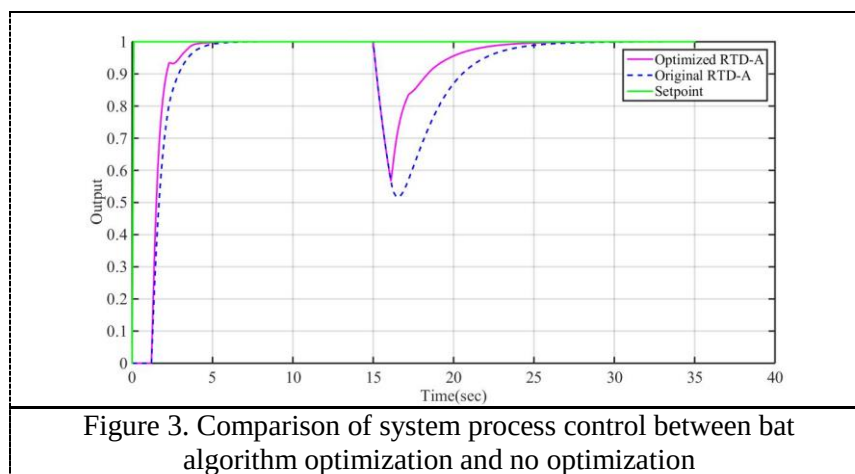


Figure 3. Comparison of system process control between bat algorithm optimization and no optimization

As shown in the figure above, in terms of setpoint tracking and interference suppression, the optimized control effect is significantly better than no optimization. The optimized system responds more quickly to the step, has stronger interference suppression capability, no overshoot, and the system has better robust performance.

4. conclusion

Through this simulation experiment, it is proved that the RTD-A controller optimized by the empirical bat algorithm has better control performance than the RTD-A controller without algorithm optimization. Specifically, the response speed to the step becomes faster, the suppression ability of the interference becomes stronger, there is no overshoot, and the robustness of the system becomes better. Although the four parameters of the RTD-A controller are clear, they cannot be excluded from each other. Based on this, this paper adopts the offline intelligent parameter optimization for the control process with set point tracking and interference suppression within a limited time step. The obtained simulation results also further illustrate the desirability of offline parameter tuning. The reliability of the performance function selection also provides a new idea for the controller to off-line intelligent parameter tuning.

Acknowledgment

The work of this paper is supported by the National Natural Science Foundation of China (Grant No. 21676012) and the Fundamental Research Funds for the Central Universities (Project XK1802-4).

References

- [1] Ogunnaike, B.A., Mukati, K. (2006) An alternative structure for the next generation regulatory controllers: Part I: Basic theory for design, development and implementation. Journal of

- Process Control.,16(15):499-509.
- [2] Danielle, D., Douglas, J.C. (2003) Tuning Guidelines of a Dynamic Matrix Controller for Integrating (Non-Self-Regulating) Processes. American Chemical Society., 42:1739-1752.
- [3] McIntosh, A.R., Shah, S.L., Fisher, D.G. (1989) Selection of Tuning Parameters for Adaptive Generalized Predictive Control. In Proceedings of the 1989 American Control Conference. Pittsburgh.
- [4] McIntosh, A.R., Shah, S.L., Fisher, D.G. (1991) Selection of Tuning Parameters for Adaptive Generalized Predictive Control. Cam.J.Chen.Eng., 69:97-110.
- [5] Bououden, S., Chadli, M., Karimi, H.R. (2015) An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. Information Sciences., 299:143-158
- [6] Liu, W., Wang, G. (2000) Auto-Tuning Procedure for Model-Based Predictive Controller. In: Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics. Nashville, TN.