Multi-Objective Gain Optimizer for an Active Disturbance Rejection Controller

Brayden DeBoon, Brayden Kent, Maciej Lacki, Scott Nokleby, and Carlos Rossa†
Faculty of Engineering and Applied Science,
Ontario Tech University, Oshawa, Ontario, Canada
{brayden.deboon, brayden.kent, maciej.lacki, scott.nokleby, carlos.rossa}@uoit.ca

***Abstract*—Active Disturbance Rejection Control (ADRC) has proven to be an efficient control method, however, the tuning of its parameters is a complicated endeavor. This paper explores the use of reference point based dominance in the traditional multi-objective non-dominated sorting genetic algorithm (NSGA-II) to perform the parameter tuning. The algorithm is applied to a simulation and physical implementation of an inverted pendulum system. The optimization method generated values that offered suitable performance among various fronts.**

***Index Terms*—Active Disturbance Rejection Control, Genetic Algorithms, Multi-Objective Optimization**

I. Introduction

Active Disturbance Rejection Control (ADRC) is an error-based method used to control the behavior of a generic plant. ADRC has the advantage of being able to compensate for disturbances to the plant compared to other control methods such as Proportional-Integral-Derivative (PID) [1]. Generally, a PID controller is tuned for a specific operation, where the disturbance introduced to a plant is constant or negligible. This may be sufficient for many cases, however, if the process is sensitive to control effort or significant and/or random disturbances are experienced, a more robust control method should be used.

Robust controllers are often model-based. This adds an element of complexity to the controller design and requires significantly more background knowledge about a plant to create the model. In many cases creating a model for a plant is not feasible or, if time is of the essence, resource consuming. This is where active disturbance rejection flourishes, since ADRC is error based and the exact mathematical model need not be known. ADRC is a viable substitute for PID where a more robust controller is necessary [2]–[5]. PID controllers have three tuning parameters, each with well defined properties. ADRC, however, can have upwards of seven tuning parameters. Genetic Algorithms (GAs) were used to optimize an ADRC for an unmanned underwater vehicle [6] and for an aircraft [7]. Particle Swarm Optimization and their variants were used in the design of force controllers [8], temperature control [9], and rocket position [10]. In other applications Ant Colony Optimization [11] and a Chaotic Cloud Cloning Selection Algorithm [12] were also used. All of these methods used single objective optimization algorithms to optimize only one specific parameter of their designs. Optimizing physical systems, however, is not a single objective task. An ADRC can be used to optimize conflicting parameters such as rise time, settling time, overshoot, controller effort, and tracking error. In the majority of design problems these objectives need to be considered and balanced.

A better approach to automate the tuning of an ADRC should incorporate a multi-objective optimizer. Standard algorithms used to solve multi-objective problems include NSGA-II, SPEA2, and NCRO. However, in problems with many objectives the performance of these algorithms drops drastically [13], [14]. For this reason, other solvers capable of solving multi-objective problems have to be used.

The problem encountered by all GA based solvers can be traced back to the dominance of points in multi-objective problems. As the number of objectives increases the number of non-dominated solutions also increases. With enough objectives, all points in the solution become non-dominated. To address this issue, researchers have suggested use of reference point domination. In NSGA-III reference-point dominance is used to improve the diversity of the solutions along the Pareto front [15]. The algorithm forces the solutions to distribute along the searchspace, which can guarantee that solutions will be found relatively fast [16]. This concept was further developed in [17], where another algorithm, θ-NSGA-III, used the same reference points in NSGA-III to push solutions closer to the Pareto front. This method was than combined with preference incorporation approaches in [18] to create a new algorithm, RPD-NSGA-II. This algorithm further improved convergence and diversity of the solutions while outcompeting both of its predecessors. Since the RPD-NSGA-II algorithm was the most efficient of the existing multi-objective solvers, it was selected to tackle the multi-objective problem presented in tuning the variables present in an ADR controller.

II. Plant Response Identification

For the purpose of this paper, the plant to be evaluated will be a classical inverted pendulum on a cart. A simplified drawing of the system is shown in Fig. 1. The cart with mass M is moved along a linear rail via a timing belt actuated by

† Corresponding author.
We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding number 2018- 06074]. Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [numéro de référence 2018-06074].

978-1-7281-2723-1/19/$31.00 ©2019 IEEE

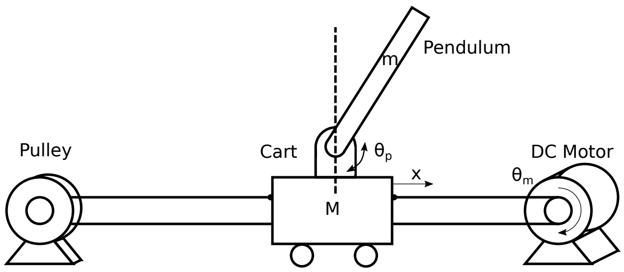

Fig. 1. Simplified setup of an inverted pendulum on a cart.

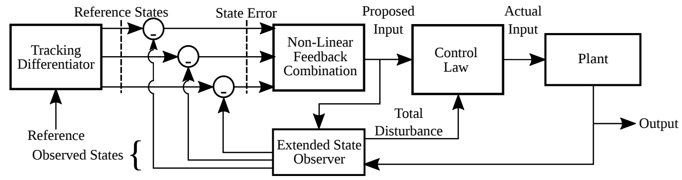

Fig. 2. Block diagram of the controller. The tracking differentiator creates a transient profile of the states to compensate for and limit drastic changes caused by differentiation; the nonlinear feedback creates a weighted sum of all state error and proposes an input to the plant. The extended state observer creates a value for total disturbance by comparing the actual output with an estimated output.

a DC motor. The equations of motion about the cart in the horizontal direction x can be summarized as follows:

$$(M+m)\ddot{x}+b_c\dot{x}+ml\ddot{\theta}_p cos(\theta_p)-ml\dot{\theta}_p^2 sin(\theta_p)=\frac{2\tau_m}{d_p}, \quad (1)$$

where m is the mass of the pendulum, b_c is the viscous friction between the cart and the linear rail, l is the distance between the pivot point and the pendulum mass centre, τ_m is the motor torque, d_p is the pitch diameter of the timing belt pulley, and θ_p is the angular position of the pendulum. Throughout this paper the operators $\dot{(\cdot)}$ and $\ddot{(\cdot)}$ represent the first and second time derivatives, respectively. The cart linear acceleration $\ddot{x}$ can be equated to the motor shaft angular acceleration $\ddot{\theta}_m$ through the timing belt pulley's pitch diameter:

$$\ddot{x}=\frac{d_p}{2}\ddot{\theta}_m, \quad (2)$$

and the motor torque τ_m can be related to the control input (voltage V_m) through the relationship:

$$\tau_m=\frac{K_M}{R_a}V_m, \quad (3)$$

where K_M is the motor constant and R_a is the motor winding resistance. Considering the forces acting normal to the pendulum, the following can be obtained:

$$(J_p+ml^2)\ddot{\theta}_p+mglsin(\theta_p)+ml\ddot{x}cos(\theta_p)=0, \quad (4)$$

where J_p is the pendulum inertia, and g is the gravitational constant. Since the pendulum will attempt to be controlled around the π radians position, the small angle approximation for deviation angle θ_{dev}, $(cos(\theta_p)\approx -1, sin(\theta_p)=sin(\pi-\theta_{dev})\approx -\theta_{dev})$ is used to approximate the above equation along with $\dot{\theta}_p^2\approx 0$ to provide the resulting equations of motion for Eqs. (1) and (4), respectively, as:

$$(M+m)\ddot{x}+b_c\dot{x}-ml\ddot{\theta}_{dev}=\frac{2\tau_m}{d_p}, \quad (5)$$

$$(J_p+ml^2)\ddot{\theta}_{dev}+mgl\theta_{dev}-ml\ddot{x}=0, \quad (6)$$

Combining Eqs. (2), (3), (5), and (6) yields:

$$\ddot{\theta}_{dev}=\frac{(M+m)mlg}{q}\theta_{dev}-\frac{mlb_cd_p}{2q}\dot{\theta}_m+\frac{2mlK_M}{R_aq}V_m \quad (7)$$

$$\ddot{\theta}_m=\frac{2m^2l^2g}{d_pq}\theta_{dev}-\frac{J_p+ml^2}{q}\dot{\theta}_m+\frac{2K_M(J_p+ml^2)}{d_pR_aq}V_m \quad (8)$$

where $q=(M+m)J_p+Mml^2$. With the above equations, the state space model is formed as follows:

$$\begin{bmatrix}\dot{\theta}_m\\ \ddot{\theta}_m\\ \dot{\theta}_{dev}\\ \ddot{\theta}_{dev}\end{bmatrix}=\begin{bmatrix}0&1&0&0\\0&a_{22}&a_{23}&0\\0&0&0&1\\0&a_{42}&a_{43}&0\end{bmatrix}\begin{bmatrix}\theta_m\\ \dot{\theta}_m\\ \theta_{dev}\\ \dot{\theta}_{dev}\end{bmatrix}+\begin{bmatrix}0\\b_2\\0\\b_4\end{bmatrix}V_m \quad (9)$$

where:

$$a_{22}=\frac{mlb_cd_p}{2q},\ a_{23}=\frac{(M+m)mlg}{q},\ a_{42}=\frac{J_p+ml^2}{q},$$
$$a_{43}=\frac{2m^2l^2g}{d_pq},\ b_2=\frac{2mlK_M}{R_aq},\ b_4=\frac{2K_M(J_p+ml^2)}{d_pR_aq}.$$

III. Generalized Multi-State Active Disturbance Rejection Controller

A block diagram of the controller is shown in Fig. 2. It shows a system with three states, however, the number of states is a variable component of the controller. Consider a SISO time varying system with n states $x_i\in R$, $i=1,2,..,n$, an extension of [19]. The system can be described as:

$$\begin{cases}\dot{x}_1=f_1(t,x_1,x_2,\ldots,x_n,D(t))\\ \dot{x}_2=f_2(t,x_1,x_2,\ldots,x_n,D(t))\\ \vdots\\ \dot{x}_n=f_n(t,x_1,x_2,\ldots,x_n,D(t))+b(t,x_1,x_2,\ldots,x_n)u\\ y=x_1,\end{cases} \quad (10)$$

where $f_i, i=1,2,\ldots,n$ and b are non-linear functions representing the system, including external disturbance $D(t)$. $u(t)$ is the control input and $y(t)$ is the output. Although there are n 'total disturbance' terms $f_i, i=1,2,\ldots,n$, one can estimate all disturbances by setting $\bar{x}_1=y$ and:

$$\bar{x}_2=f_1(t,x_1,x_2,\ldots,x_n,D(t)), \quad (11)$$

Therefore, Eq. (10) can be written as:

$$\begin{cases} \dot{\bar{x}}_1 = \bar{x}_2 \\ \dot{\bar{x}}_2 = \bar{x}_3 = \frac{\partial \bar{x}_2}{\partial t} + \frac{\partial \bar{x}_2}{\partial x_1}\bar{x}_2 + \frac{\partial \bar{x}_2}{\partial D}\frac{\partial D}{\partial t} \\ \quad + \frac{\partial \bar{x}_2}{\partial x_2} f_2(t, x_1, x_2, \ldots, x_n, D(t)) + \sum_{k=3}^{n} \frac{\partial \bar{x}_2}{\partial x_k}\bar{x}_2 \\ \vdots \\ \dot{\bar{x}}_n = \frac{\partial \bar{x}_n}{\partial t} + \sum_{k=1}^{n-1} \frac{\partial \bar{x}_n}{\partial x_k}\bar{x}_n + \frac{\partial \bar{x}_n}{\partial D}\frac{\partial D}{\partial t} \\ \quad + \frac{\partial \bar{x}_n}{\partial x_n} f_n(t, x_1, x_2, \ldots, x_n, D(t)) \\ \quad + \frac{\partial \bar{x}_n}{\partial x_n} b(t, x_1, x_2, \ldots, x_n)u \\ y = \bar{x}_1, \end{cases} \tag{12}$$

To estimate the total disturbance, consider first a linear estimation for the potentially non-linear b term as $\bar{b}(t)$. Extending the system to consider a new state representing total disturbance as $\bar{x}_{n+1}$, $\dot{\bar{x}}_n$ and $\dot{\bar{x}}_{n+1}$ in (12) can be redefined as:

$$\begin{cases} \dot{\bar{x}}_n = \bar{x}_{n+1} + \bar{b}(t)u(t) \\ \dot{\bar{x}}_{n+1} = \bar{x}_n \end{cases} \tag{13}$$

where the total disturbance can be combined into:

$$\begin{aligned} \bar{x}_{n+1} = & \frac{\partial \bar{x}_n}{\partial t} + \sum_{k=1}^{n-1} \frac{\partial \bar{x}_n}{\partial x_k}\bar{x}_n + \frac{\partial \bar{x}_n}{\partial D}\frac{\partial D}{\partial t} \\ & + \frac{\partial \bar{x}_n}{\partial x_n} f_n(t, x_1, x_2, \ldots, x_n, D(t)) \\ & + \left(\frac{\partial \bar{x}_n}{\partial x_n} b(t, x_1, x_2, \ldots, x_n) - \bar{b}(t) \right) u(t) \end{aligned} \tag{14}$$

Part of the ADRC scheme implements an extended state observer. With the system of extended states, one can write the state estimator using a Luenberger observer as follows:

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 + \beta_{01}(y(t) - \hat{y}(t)) \\ \dot{\hat{x}}_2 = \hat{x}_3 + \beta_{02}(y(t) - \hat{y}(t)) \\ \vdots \\ \dot{\hat{x}}_n = \hat{x}_{n+1} + \beta_n(y(t) - \hat{y}(t)) + \bar{b}(t)u(t) \\ \dot{\hat{x}}_{n+1} = \beta_{n+1}(y(t) - \hat{y}(t)) \\ \hat{y} = \hat{x}_1, \end{cases} \tag{15}$$

where $\beta_0 i$ $i = 1, 2, \ldots, n+1$ are the observer gains for the general system. This allows the ADR control law to be:

$$u = -\frac{\hat{x}_{n+1} - u_0}{\bar{b}} \tag{16}$$

where u_0 is the proposed input from a non-linear feedback weighted combiner:

$$u_0 = k_1(r - \hat{x}_1) + k_2(\dot{r} - \hat{x}_2) + \cdots + k_n(\overset{n-1}{r} - \hat{x}_n) + \overset{n}{r}, \tag{17}$$

where r is the reference control input and k_i, $i = 1, 2, \ldots, n$ are tunable gains to achieve a desired performance depending on the needs of the user. The encased equations for the state estimator, Eq. (15), represents a particular state class and its derivatives. For instance, this could include linear position, linear velocity, and acceleration. Additional systems of equations can be written for other state classes as in [20].

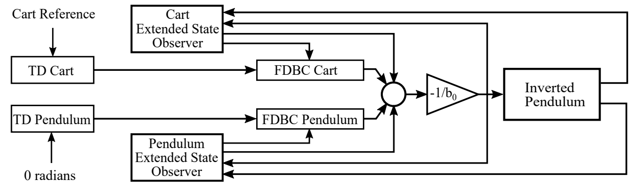

Fig. 3. Controller Schematic for the Inverted Pendulum

A. Controller Gains

Throughout this paper, the discrete nonlinear feedback combiner will take the form of the $fhan$ function from [21], which has three extra controllable parameters per function call. The resulting control law for the inverted pendulum is:

$$u_0 = -\frac{fhan(e_1, ce_2, h_1, r_1) - z_3}{\bar{b}}, \tag{18}$$

where e_1 and e_2 are the observed proportional and time-varying error values, c is a fine tuning parameter, h_1 is a controller aggressiveness factor closely related to the sampling frequency of the controller, r_1 is an acceleration limiting function, and z_3 is the total disturbance estimated by the observer. It is important to note that these three gains are unique to every controlled state in the plant that is observed by an extended state observer. Taking all of the total controller tunable parameters into consideration, the gains that are to be optimized in this paper are the observer gains β_i, $i = 1, 2, \ldots, n+1$ from Eq. (15) and the discrete feedback combiner gains c, h_1, and r_1 from Eq. (18). Throughout the literature on active disturbance rejection control, the choices of the aforementioned gains are more objective than analytical. The proposed gains from [21] are at most a good starting point for an optimization process, which will be used to obtain a set of gains for any unique low-order plant. The complete block diagram for the proposed controller is shown in Fig. 3.

IV. Controller Gain Optimization

The performance of the ADRC system will depend on the values of the gains used. For the inverted pendulum plant there are two extended state observers as shown in Fig. 3: one that monitors the position of the cart and the other that monitors the angular position of the pendulum. Each extended state observer has three observer gains, β_{01}, β_{02}, and β_{03}, an acceleration limit of the tracking differentiator, r, and the damping coefficient, c. Lastly, h_{1p} and h_{1c} are parameters for the pendulum and cart, respectively, that determine the aggressiveness of their control loops [21]. As previously discussed, the gains will be adjusted to optimize performance for several objectives, namely: tracking error of the cart, tracking error of the pendulum, controller effort, rise time, percent overshoot, settling time, and steady state error as shown in Fig. 4. Repeated sampling was performed to account for the stochastic nature of the algorithm. Each sample was performed with 1000 generations. The crossover and mutation variation parameters were set to 20, as per recommendation from [22]. Boundary conditions were applied to the parameters to prevent

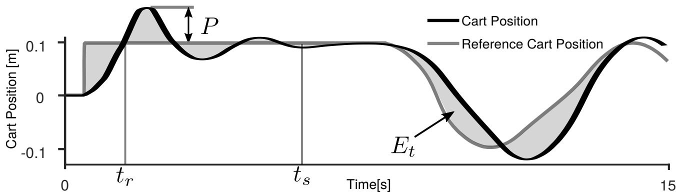

Fig. 4. The carts position (thick black line) changes in response to the reference signal (thick grey line). Rise time, t_r, percent overshoot, P, and settling time, t_s, within 5% error margin. The tracking error, E_t, is the integral of the error between the reference and the response signal (shaded regions).

TABLE I
OPTIMIZER BOUNDARY CONDITIONS

Variable	Range
Aggr. Control Loop	$1e^{-5} < h_{1p},\ h_{1c} < 1e^{-1}$
Acceleration Limit	$1e^{-3} < r_p,\ r_c < 1e^{2}$
Damping Coefficient	$0.5 < c_p,\ c_c < 1.5$
Viscous Friction Est.	$1e^{-5} < b_0 < 1e^{-1}$
Observer Gains	$0.5 < \beta_{01p},\ \beta_{01c} < 2$
	$1e^{-8} < \beta_{0(2,3)p},\ \beta_{0(2,3)c} < 1e^{-3}$

infeasible solutions. The conditions are presented in Table I. In order to simulate a rather challenging environment, the inverted pendulum was commanded to respond to a change in desired cart position, starting with a step input from 0 m to 0.1 m, and then following a sine wave with angular frequency 1 rad/s as $0.1\sin(t)$. Apart from the change in cart position, a disturbance was added to the system to try and bring the pendulum out of equilibrium. An impulse of 0.087 radians ($\approx 5^o$) applied to the pendulum angle was introduced to the system at 5.5 seconds to show that the rejection controller is able to handle this disturbance.

As discussed in the introduction, the optimizer used for this task is the RPD-NSGA-II from [18]. The reference normalized hyperplane was set to have 10 divisions between fitness values. From this hyperplane, each normalized candidate fitness vector was given a convergence distance d_1 to its nearest reference point using the diversity metric d_2. The density of each reference point was then computed and used to truncate the final reference-point dominated front to be passed to the next generation. The crossover scheme used in this optimization process was a simulated binary crossover (SBX) which is evaluated in more depth in [23] and [24]. The mutation variation was done with polynomial mutation similar to that described in [22] and the variation process was inherited from [25].

V. RESULTS

To verify the results of the optimizer and simulation, a set of the tuned gains were implemented on a physical inverted pendulum system. The angle of the pendulum bar and the rotation of the motor shaft were measured using AMT102-V quadrature encoders. Fig. 5 demonstrates the performace of the optimized parameters at various generations by selecting members of the population that dominate on the tracking error

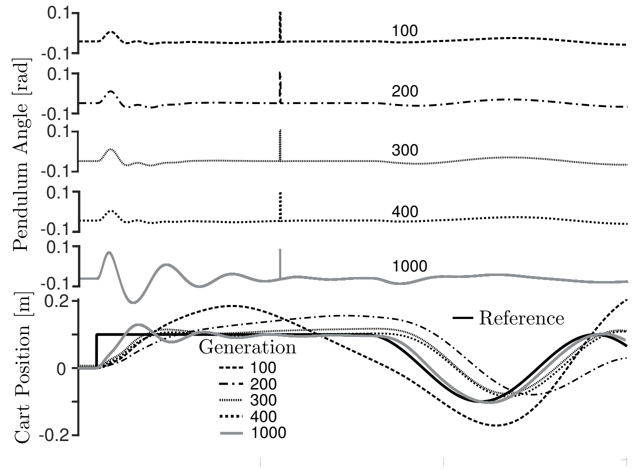

Fig. 5. Simulated optimization results with emphasis on tracking error

front. Fig. 6 demonstrates the experimentally validated cart performance results with the ADRC gain values found with the optimizer.

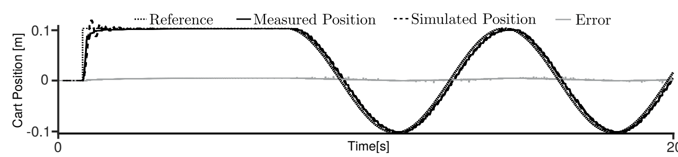

Fig. 6. Experimental results. The inverted pendulum model validation for the cart position with the pendulum detached.

The RPD-NSGA-II algorithm was able to compute the parameter values to achieve the desired performance on the inverted pendulum system. The simulation showed a promising time response to the set of optimized parameters. These values were also tested on the physical implementation of the inverted pendulum and were also successful.

VI. CONCLUSIONS

The most difficult portion of ADRC is the tuning of the system parameters. Presented in this paper was the successful implementation of RPD-NSGA-II to optimize the parameters required for ADRC on an inverted pendulum system. By using a multi-objective optimization technique coupled with a simulation, the parameters required to achieve the desired performance for a physical system were determined. The RPD-NSGA-II routine proved capable of handling this multi-objective optimization problem to provide the end user with a set of dominating solutions such that the user can choose gain values based on their needs. There are a wide number of applications for an ADRC optimization method. Testing the method on more complicated systems with an increased number of objectives could further validate the robustness and applicability of the optimizer.

REFERENCES

[1] M. Przybyła, M. Kordasz, R. Madoński, P. Herman, and P. Sauer, "Active disturbance rejection control of a 2dof manipulator with significant modeling uncertainty," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 60, no. 3, pp. 509–520, 2012.

[2] G. Wu, L. Sun, and K. Y. Lee, "Disturbance rejection control of a fuel cell power plant in a grid-connected system," *Control Engineering Practice*, vol. 60, pp. 183–192, 2017.

[3] B. Ahi and A. Nobakhti, "Hardware implementation of an adrc controller on a gimbal mechanism," *IEEE Transactions on Control Systems Technology*, 2017.

[4] H.-L. Xing, J.-H. Jeon, K. Park, and I.-K. Oh, "Active disturbance rejection control for precise position tracking of ionic polymer–metal composite actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 86–95, 2013.

[5] Z.-L. Zhao and B.-Z. Guo, "On active disturbance rejection control for nonlinear systems using time-varying gain," *European Journal of Control*, vol. 23, pp. 62–70, 2015.

[6] K. Hu, X.-f. Zhang, and C.-b. Liu, "Unmanned underwater vehicle depth adrc based on genetic algorithm near surface," *Acta Armamentarii*, vol. 34, no. 2, pp. 217–222, 2013.

[7] H. Geng, H. Yang, Y. Zhang, and H. Chen, "Auto-disturbances-rejection controller design and it's parameter optimization for aircraft longitudinal attitude," *Journal of System Simulation*, vol. 22, no. 1, pp. 89–91, 2010.

[8] F. Li, Z. Zhang, A. Armaou, Y. Xue, S. Zhou, and Y. Zhou, "Study on adrc parameter optimization using cpso for clamping force control system," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[9] G. Hou, M. Wang, L. Gong, and J. Zhang, "Parameters optimization of adrc based on adaptive cpso algorithm and its application in main-steam temperature control system," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 497–501.

[10] R.-l. Wang, B.-c. Lu, Y.-l. Hou, and Q. Gao, "Passivity-based control for rocket launcher position servo system based on adrc optimized by ipso-bp algorithm," *Shock and Vibration*, vol. 2018, 2018.

[11] Z. Yin, C. Du, J. Liu, X. Sun, and Y. Zhong, "Research on autodisturbance-rejection control of induction motors based on an ant colony optimization algorithm," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3077–3094, 2018.

[12] Y. Zhang, C. Fan, F. Zhao, Z. Ai, and Z. Gong, "Parameter tuning of adrc and its application based on cccsa," *Nonlinear dynamics*, vol. 76, no. 2, pp. 1185–1194, 2014.

[13] D. Kalyanmoy *et al.*, *Multi objective optimization using evolutionary algorithms*. John Wiley and Sons, 2001.

[14] K. Deb and D. Saxena, "Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, 2006, pp. 3352–3360.

[15] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints." *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[16] G. C. Ciro, F. Dugardin, F. Yalaoui, and R. Kelly, "A nsga-ii and nsga-iii comparison for solving an open shop scheduling problem with resource constraints," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1272–1277, 2016.

[17] Y. Yuan, H. Xu, and B. Wang, "An improved nsga-iii procedure for evolutionary many-objective optimization," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 2014, pp. 661–668.

[18] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y.-S. Ong, "A new decomposition-based nsga-ii for many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1191–1210, 2018.

[19] Y. Huang and W. Xue, "Active disturbance rejection control: methodology and theoretical analysis," *ISA transactions*, vol. 53, no. 4, pp. 963–976, 2014.

[20] Y. Long, Z. Du, L. Cong, W. Wang, Z. Zhang, and W. Dong, "Active disturbance rejection control based human gait tracking for lower extremity rehabilitation exoskeleton," *ISA transactions*, vol. 67, pp. 389–397, 2017.

[21] J. Han, "From pid to active disturbance rejection control," *IEEE transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[22] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 1, pp. 1–28, 2014.

[23] K. Kumar, "Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems," *Complex Syst*, vol. 9, pp. 431–454, 1995.

[24] R. B. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[25] Z. Zhao, B. Liu, C. Zhang, and H. Liu, "An improved adaptive nsga-ii with multi-population algorithm," *Applied Intelligence*, pp. 1–12, 2018.

View publication stats