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 out-competing 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
a DC motor. The equations of motion about the cart in the horizontal direction \( x \) can be summarized as follows:
\[
(M + m) \ddot{x} + b_c \dot{x} + m l \dot{\theta}_p \cos(\theta_p) - m l \dot{\theta}_p^2 \sin(\theta_p) = \frac{2 \tau_m}{d_p},
\]  
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, \( \tau_m \) is the motor torque, \( d_p \) is the pitch diameter of the timing belt pulley, and \( \theta_p \) is the angular position of the pendulum. Throughout this paper the operators \( (\cdot) \) and \( (\cdot) \) represent the first and second time derivatives, respectively. The cart linear acceleration \( \ddot{x} \) can be equated to the motor shaft angular acceleration \( \dot{\theta}_m \) through the timing belt pulley’s pitch diameter:
\[
\ddot{x} = \frac{d_p}{\dot{\theta}_m},
\]  
and the motor torque \( \tau_m \) can be related to the control input (voltage \( V_m \)) through the relationship:
\[
\tau_m = \frac{K_M}{R_a} V_m, 
\]  
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:
\[
(J_p + m l^2) \ddot{\theta}_p + m gl \sin(\theta_p) + m l \dot{\theta}_p \dot{\theta}_p \cos(\theta_p) = 0, 
\]  
where \( J_p \) is the pendulum inertia, and \( g \) is the gravitational constant. Since the pendulum will attempt to be controlled around the \( \pi \) radians position, the small angle approximation for deviation angle \( \theta_{dev} \), \( \cos(\theta_p) \approx -1, \sin(\theta_p) \approx \sin(\pi - \theta_{dev}) \approx \theta_{dev} \approx 0 \) 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:
\[
(M + m) \ddot{x} + b_c \dot{x} + m l \ddot{\theta}_p = \frac{2 \tau_m}{d_p},
\]
\[
(J_p + m l^2) \ddot{\theta}_p + m gl \theta_p + m l \dot{\theta}_p \dot{\theta}_p = 0.
\]
Combining Eqs. (2), (3), (5), and (6) yields:
\[
\ddot{\theta}_{dev} = \frac{(M + m) ml d_p \ddot{\theta}_{dev} - m l b_c d_p \dot{\theta}_m + 2 m l K_M}{q} V_m,
\]
\[
\dot{\theta}_m = \frac{2 m l^2 g}{d_p q} \theta_{dev} - \frac{J_p + m l^2}{q} \theta_m + \frac{2 K_M (J_p + m l^2)}{d_p R_a q} V_m.
\]

where \( q = (M + m) J_p + M ml^2 \). With the above equations, the state space model is formed as follows:
\[
\begin{bmatrix}
\dot{\theta}_m \\
\dot{\theta}_{dev} \\
\dot{\theta}_{dev} \\
\theta_m
\end{bmatrix}
= 
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & a_{22} & a_{23} & 0 \\
0 & 0 & 0 & 1 \\
0 & a_{42} & a_{43} & 0
\end{bmatrix}
\begin{bmatrix}
\theta_m \\
\theta_{dev} \\
\theta_{dev} \\
\theta_m
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
b_2 \\
b_4
\end{bmatrix}
V_m
\]
\]  
where:
\[
a_{22} = \frac{m l b_c d_p}{q},
\]
\[
a_{23} = \frac{(M + m) ml g}{q},
\]
\[
a_{42} = \frac{J_p + m l^2}{q},
\]
\[
a_{43} = \frac{2 m l^2 g}{d_p q},
\]
\[
b_2 = \frac{2 m l K_M}{R_a q},
\]
\[
b_4 = \frac{2 K_M (J_p + m l^2)}{d_p R_a q}.
\]

### 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 \( x_i \in R, i = 1, 2, \ldots, n \), an extension of [19]. The system can be described as:
\[
\begin{aligned}
\dot{x}_1 &= f_1(t, x_1, x_2, \ldots, x_n, D(t)) \\
\dot{x}_2 &= f_2(t, x_1, x_2, \ldots, x_n, D(t)) \\
&\quad \vdots \\
\dot{x}_n &= f_n(t, x_1, x_2, \ldots, x_n, D(t)) + b(t, x_1, x_2, \ldots, x_n) u \\
y &= x_1,
\end{aligned}
\]
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 \( \ddot{x}_1 = y \) and:
\[
\ddot{x}_2 = f_1(t, x_1, x_2, \ldots, x_n, D(t)), 
\]  
\[
\ddot{x}_3 = f_2(t, x_1, x_2, \ldots, x_n, D(t)),
\]  
\[
\ddot{x}_4 = f_3(t, x_1, x_2, \ldots, x_n, D(t)),
\]  
\[
\ddot{x}_n = f_n(t, x_1, x_2, \ldots, x_n, D(t)) + b(t, x_1, x_2, \ldots, x_n) u 
\]
Therefore, Eq. (10) can be written as:

\[
\begin{align*}
\dot{x}_1 &= \dot{x}_2 \\
\dot{x}_2 &= \dot{x}_3 - \frac{\partial x_2}{\partial t} + \frac{\partial x_3}{\partial t} \frac{\partial D}{\partial t} + \sum_{k=1}^{n} \frac{\partial x_2}{\partial x_k} f_k(t, x_1, x_2, \ldots, x_n, D(t)) + \sum_{k=1}^{n} \frac{\partial x_3}{\partial x_k} \dot{x}_k \\
\vdots \\
\dot{x}_n &= \frac{\partial x_n}{\partial t} + \sum_{k=1}^{n} \frac{\partial x_n}{\partial x_k} \dot{x}_k + \frac{\partial x_n}{\partial D} \frac{\partial D}{\partial t} + \frac{\partial x_n}{\partial x_n} f_n(t, x_1, x_2, \ldots, x_n, D(t)) + \frac{\partial x_n}{\partial x_n} b(t, x_1, x_2, \ldots, x_n) u(t) + \sum_{k=1}^{n} \frac{\partial x_n}{\partial x_k} \dot{x}_k
\end{align*}
\]

(12)

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

\[
\begin{align*}
\dot{x}_n &= \tilde{x}_{n+1} + \tilde{b}(t) u(t) \\
\dot{x}_{n+1} &= \tilde{x}_n
\end{align*}
\]

(13)

where the total disturbance can be combined into:

\[
\tilde{x}_{n+1} = \frac{\partial \tilde{x}_n}{\partial t} + \sum_{k=1}^{n} \frac{\partial \tilde{x}_n}{\partial x_k} \dot{x}_k + \frac{\partial \tilde{x}_n}{\partial D} \frac{\partial D}{\partial t} + \frac{\partial \tilde{x}_n}{\partial x_n} f_n(t, x_1, x_2, \ldots, x_n, D(t)) + \frac{\partial \tilde{x}_n}{\partial x_n} b(t, x_1, x_2, \ldots, x_n) u(t)
\]

(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:

\[
\begin{align*}
\dot{\hat{x}}_1 &= \hat{x}_2 + \beta_{01}(y(t) - \hat{y}(t)) \\
\dot{\hat{x}}_2 &= \hat{x}_3 + \beta_{02}(y(t) - \hat{y}(t)) \\
\vdots \\
\dot{\hat{x}}_n &= \hat{x}_{n+1} + \beta_n(y(t) - \hat{y}(t)) + \tilde{b}(t) u(t) \\
\dot{\hat{x}}_{n+1} &= \beta_{n+1}(y(t) - \hat{y}(t)) \\
\dot{\hat{y}} &= \hat{x}_1,
\end{align*}
\]

(15)

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

\[
u = -\frac{\dot{x}_{n+1} - u_0}{b}
\]

(16)

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

\[
u_0 = k_1(r - \hat{x}_1) + k_2(\hat{r} - \hat{x}_2) + \ldots + k_n(n-r - \hat{x}_n) + \hat{r}
\]

(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].
TABLE I

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggr. Control Loop</td>
<td>$1e^{-5} &lt; h_{1p}, h_{1c} &lt; 1e^{-1}$</td>
</tr>
<tr>
<td>Acceleration Limit</td>
<td>$1e^{-3} &lt; r_p, r_c &lt; 1e^{2}$</td>
</tr>
<tr>
<td>Damping Coefficient</td>
<td>$0.5 &lt; c_p, c_c &lt; 1.5$</td>
</tr>
<tr>
<td>Viscous Friction Est.</td>
<td>$1e^{-5} &lt; b_0 &lt; 1e^{-1}$</td>
</tr>
<tr>
<td>Observer Gains</td>
<td>$0.5 &lt; \beta_{01p}, \beta_{01c} &lt; 2$</td>
</tr>
<tr>
<td></td>
<td>$1e^{-8} &lt; \beta_{0(2,3)p}, \beta_{0(2,3)c} &lt; 1e^{-3}$</td>
</tr>
</tbody>
</table>
REFERENCES


