Control Engineering Practice 109 (2021) 104733

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

# Multi-objective gain optimizer for a multi-input active disturbance rejection controller: Application to series elastic actuators ☆

Brayden DeBoon, Scott Nokleby, Carlos Rossa *

*Faculty of Engineering and Applied Science, Ontario Tech University, Oshawa, Ontario, Canada*

ARTICLE INFO

*Keywords:*
Active disturbance rejection
Series elastic actuator
Multi-objective control

ABSTRACT

Series elastic actuators (SEA) have been gaining increasing popularity as a mechanical drive in contemporary force-controlled robotic manipulators thanks to their ability to infer the applied torque from measurements of the elastic element's deflection. Accurate deflection control is crucial to achieve a desired output torque and, therefore, unmodelled dynamics and dynamic loads can severely compromise force fidelity. Multi-input active disturbance rejection controllers (ADRC) have the ability to estimate such disturbances affecting the plant behaviour and cancel them via an appropriate feedback controller. Thus, they offer a promising control architecture for SEA. ADRC, however, can have upwards of eight tuning parameters for each controlled state. Tuning the controller becomes quite challenging, especially in the context of multi-input, multi-objective control.

This paper tackles the problem of ADRC tuning as a multi-parametric and multi-objective optimization approach. An ADRC is developed to regulate the output torque of a multi-input hybrid motor-brake–clutch SEA. The controller has a total of 22 tunable parameters. Point dominance-based nondominated sorting genetic algorithm is used to find the optimal control gains, first considering nine individual control objectives, and then in the context of multi-objective. The algorithm provides a set of potential solutions that highlight the tradeoffs between the control objectives. It is up to the discretion of the designer to select the appropriate solution that best suits a given application. The approach is validated experimentally and the results are compared with a simulated model. Experimental results confirm the suitability of the proposed approach for single and multiple control objectives in a variety of experimental scenarios and show good agreement with the analytical model.

## 1. Introduction

Series elastic actuators (SEAs) with passive compliance are gaining terrain in the field of collaborative human–robot systems. They incorporate an elastic element between the mechanical drive and the robot arm, trading off bandwidth for gains in stability, force control, and robustness against shock loads (DeBoon, Nokleby, La Delfa, & Rossa, 2019). As a result of the inherent increasing complexity in the actuation system, SEA control architecture must be adapted to take into account unmodelled dynamics and other disturbances.

SEA often use model-based force controllers, suggesting that the controller has significant background knowledge about the plant. In many cases such as in the context of human–robot interaction, creating the model is not feasible or, if time is of the essence, resource consuming. This is where active disturbance rejection controllers (ADRC) flourish, since they are error based and the exact plant model need not be known. ADRC is a viable substitute for the familiar proportional–integral–derivative (PID) controller where a more robust control strategy is necessary (Ahi & Nobakhti, 2017; Wu, Sun, & Lee, 2017; Xing, Jeon, Park, & Oh, 2013; Zhao & Guo, 2015). PID controllers have three tuning parameters, each with well defined properties. ADRC, however, can have upwards of eight tuning parameters for each controlled state. Therefore, tuning the controller becomes challenging and depends completely on the control objective for a given application.

Single-objective genetic algorithms (GAs) and other stochastic methods have been used to tune ADRC gains in a variety of applications ranging from force and temperature control to rocket position control (Geng, Yang, Zhang, & Chen, 2010; Hou, Wang, Gong, & Zhang, 2018; Hu, Zhang, & Liu, 2013; Li et al., 2018; Wang, Lu, Hou, & Gao, 2018; Yin, Du, Liu, Sun, & Zhong, 2018; Zhang, Fan, Zhao, Ai, & Gong, 2014). Many optimization and convergence studies have been conducted on tuning the tracking differentiator and extended state observer gains for ADRCs (Du et al., 2018; Wang, Zu, Duan, & Li,

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

* Corresponding author.
*E-mail addresses:* brayden.deboon@ontariotechu.ca (B. DeBoon), scott.nokleby@ontariotechu.ca (S. Nokleby), carlos.rossa@ontariotechu.ca (C. Rossa).

https://doi.org/10.1016/j.conengprac.2021.104733
Received 4 May 2020; Received in revised form 24 November 2020; Accepted 9 January 2021
Available online xxxx

2019; Zhang, Xiao, Yu, & Xie, 2020; Zhang, Xu, & Gerada, 2019). Optimizing a controller, however, is often not a single objective task. ADRC gains have conflicting implications in the controller performance and as such, optimizing several control objectives at a time, such as rise time, settling time, overshoot, control effort, and tracking error, is impractical. In the majority of design problems and in particular in human–machine interaction systems, many of these control objectives need to be taken into account and balanced.

A more suitable approach to automate the tuning of an ADRC must incorporate multiple control objectives, giving rise to a new challenge: there may be more than a single set of parameters that satisfies the objectives (Madoński, Piosik, & Herman, 2013). Several algorithms have been developed to address this issue, including nondominated sorting genetic algorithms (NSGA-II) and strength pareto evolutionary algorithms (SPEA2). In problems with an increased number of objectives, the performance of these algorithms is known to deteriorate since behaviour becomes similar to randomly exploring the search space since most solutions are nondominated with respect to each other (Deb & Saxena, 2006; Kalyanmoy et al., 2001). For this reason, other solvers capable of solving many-objective problems are needed.

In all NSGA solvers, a solution that performs better than another in at least one objective, and not worse in any other objective is said to be dominant. In order to tune the ADRC, 9 objectives will be defined later on. With such a large number of objectives, most solutions will be uniquely optimal to a given objective and will prevent the algorithm from converging. A solution to this issue can be resolved using reference point domination to improve the diversity of solutions along the pareto front (Deb & Jain, 2014; Hernandez Mejia et al., 2017) by forcing them to distribute along the search space (Ciro, Dugardin, Yalaoui, & Kelly, 2016). This concept, called $\theta$-NSGA-III, is further expanded in Yuan, Xu, and Wang (2014) to push solutions closer to the pareto front. Preference incorporation in integrated in Elarbi, Bechikh, Gupta, Said, and Ong (2018) to create a new algorithm, the RPD-NSGA-II that further improves convergence and diversity of the solutions. This method is shown in Elarbi et al. (2018), to provide similar or better results when compared against its predecessing genetic algorithms on commonly-used benchmark problems involving up to 20 objectives. Thus, it is selected for the multi-parametric and multi-objective problem of ADRC tuning presented in this paper.

This paper tackles the problem of ADRC tuning as a multi-parametric and multi-objective optimization approach applied to SEA for human–machine interaction. An ADRC is developed to regulate the output torque of a multi-input SEA comprising of a brake and motor. Torque control is achieved indirectly by monitoring the deflection of the elastic element and adjusting it via a distributed control law for the brake and motor torques, rendering the system as a multi-input, single-output (MISO) entity. The ADRC has 22 tuning parameters, 11 for the motor and 11 for the brake controller. One of the contributions of this paper includes using the point dominance-based nondominated sorting genetic algorithm to find the optimal control gains, first for nine individual control objectives, and then for all control objectives at once. Rather than converging to a single solution, the algorithm provides a set of optimal solutions that highlight the tradeoff between different control objectives. A solution that best suits a given application can then be selected. Another contribution of this paper is the optimization of a dual input SEA controller. To the best of the authors' knowledge, this is the first implementation of a multi-parametric and multi-objective gain optimizer of ADRC gain applied specifically to a dual input SEAs. The framework proposed in this paper has applications in a variety of other control architectures and systems.

The remainder of this paper is structured as follows. First, the MISO SEA actuator is introduced and its state space model is derived. The multi-input ADRC is then laid out, including the extended state observers and the distributive control law for the brake and motor. The optimization algorithm is introduced in Section 4 along with all performance objectives, mainly, tracking error, control effort, percent overshoot, rise time, settling time, maximum input, steady-state error, disengagement time, and the number of input direction changes. Experimental results confirm the suitability of the proposed approach for single and multiple control objectives in a variety of experimental scenarios and show good agreement with the analytical model.

## 2. MISO series elastic actuator model

Consider the differentially-clutched SEA we introduced in DeBoon et al. (2019). The actuator is composed of a DC motor in series with a spring, connected to a magnetic particle brake through a differential clutch. A simplified dynamics diagram and a picture of the device are shown in Fig. 1. The actuator has three operating modes. **Mode 1:** The brake is fully engaged and the actuator acts like a classical SEA. **Mode 2**: The brake is engaged and the motor is static. The user is then directly coupled to a grounded spring whose stored energy is controlled by adjusting the braking torque. **Hybrid mode**: Both the motor and brake are engaged, and the brake and differential act as a continuous variable-slip clutch between the spring and the output. As the motor compresses the spring, the brake regulates the amount of energy stored in the spring, thereby controlling the output torque.

The actuator demonstrated above is a multi-input device that measures spring deflection to infer the output torque at the end-effector. The deflection is determined through the difference in encoder measurements between the motor-mounted encoder and the spring-side shaft encoder.

The device dynamics can be summarized by the following set of differential equations (DeBoon et al., 2019):

$$J_m\ddot{\theta}_m + b_m\dot{\theta}_m + k_s\Delta\theta_s = \tau_m \tag{1}$$

$$J_b\ddot{\theta}_b - 4b_d\dot{\theta}_u + (4b_d + b_b)\dot{\theta}_b + k_s\Delta\theta_s = \tau_b \tag{2}$$

$$J_u\ddot{\theta}_u + b_u\dot{\theta}_u + k_u\theta_u = \tau_u \tag{3}$$

where $k_s$ is the stiffness of the spring, $\theta$ is the angular position, the dot operators $\dot{(\cdot)}$ and $\ddot{(\cdot)}$ represent the first and second time derivatives, respectively. $J$ and $b$ represent the inertial and viscous friction coefficients, and $\tau$ represents a torque. Throughout this paper, subscripts $m, b, u$, and $d$ refer to the dynamics in the motor body, brake body, user/output body, and the differential gearbox, respectively. $k_u$ refers to the spring constant of an output measurement device or a passive elastic environment. $\tau_m$ is the motor torque, which for low speed can also be related through the input voltage $V_m$ of the device $\tau_m = (K_m V_m)/R_a$ where $K_m$ is the motor torque constant and $R_a$ is the winding resistance of the motor. The brake torque $\tau_b$ can be modelled as $\tau_b = f(V_b/R_b)$ where $V_b$ is the input voltage of the brake and $R_b$ is the winding resistance. This function in a natural state is nonlinear due to magnetic hysteresis of the particle brake, however, in this paper it will be simplified to be proportional to the input current through a gain $K_H$. Finally, $\tau_u$ represents the user torque acting against the satellite-side dynamics of the differential gearbox.

For an open differential layout where the torque is split evenly between the planetary gears the following equations hold (see Fig. 1a):

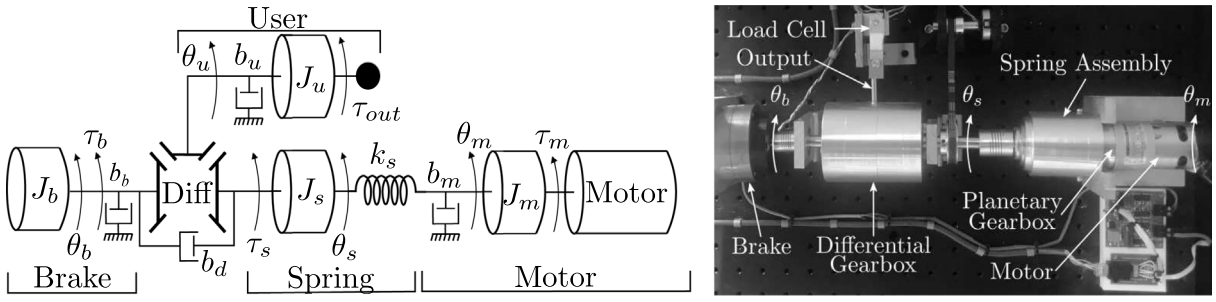
$$\frac{\tau_u}{2} = \tau_b = \tau_s \tag{4}$$

$$\theta_u = \frac{\theta_b + \theta_s}{2} \tag{5}$$

Note that (5) also holds for the first and second time derivatives, i.e. $\ddot{\theta}_u = (\ddot{\theta}_b + \ddot{\theta}_s)/2$). Combining the previous equations to make the system *independent* of the position of the user, the following two differential equations are obtained:

$$\ddot{\theta}_b = \frac{J_u\ddot{\theta}_s + (4b_d + b_u)\dot{\theta}_s + k_u\theta_s - (4b_d + 4b_b - b_u)\dot{\theta}_b + k_u\theta_b - 4\tau_b}{4J_b - J_u} \tag{6}$$

$$\ddot{\theta}_s = \frac{J_u\ddot{\theta}_b + (4b_d + b_u)\dot{\theta}_b + k_u\theta_b - (4b_d + 4b_b - b_u)\dot{\theta}_s + (k_u + 4k_s)\theta_s - 4k_s\theta_m}{4J_s - J_u} \tag{7}$$




(a) Actuator topology (b) Actuator experimental setup

**Fig. 1.** Multi-input series elastic actuator (SEA): (a) shows the SEA topology introduced in DeBoon et al. (2019). It uses a DC motor, a spring, a rheological brake, a torsion spring. The spring is located between the motor and the differential, whose other side is attached to the brake. The satellite gears of the differential make up the output shaft of the actuator; (b) is an image of the actuator showing the two inputs (motor and brake) as well as the differential clutch mechanism.

Substituting (7) into (6) yields:

$$\begin{aligned}\ddot{\theta}_b = &\frac{J_u k_s}{J}\theta_m + \frac{J_s k_u}{J}\theta_b + \frac{J_u(2b_d + b_b) - J_s(4b_d + 4b_b - b_u)}{J}\dot{\theta}_b \\ &+\frac{J_s k_u - J_u k_s}{J}\theta_s \\ &+\frac{J_s(4b_d + b_u) - J_u(b_s + 2b_d)}{J}\dot{\theta}_s + \frac{J_u - 4J_s}{J}\tau_b \end{aligned} \tag{8}$$

where $J = 4J_sJ_b - J_uJ_s - J_uJ_b$. Similarly, by inserting (6) into (7) obtains:

$$\begin{aligned}\ddot{\theta}_s = &\frac{(4J_b - J_u)k_s}{J}\theta_m + \frac{J_b k_u}{J}\theta_b + \frac{J_b(4b_d + b_u) - J_u(2b_d + b_b)}{J}\dot{\theta}_b \\ &+\frac{J_b k_u - 4J_b k_s + J_u k_s}{J}\theta_s \\ &+\frac{J_u(b_s + 2b_d) + J_b(b_u - 4b_s - 4b_b)}{J}\dot{\theta}_s - \frac{J_u}{J}\tau_b \end{aligned} \tag{9}$$

The previous equations can be rearranged to generate the multi-input state space model of the actuator for a state vector $\Theta = [\theta_m\ \dot{\theta}_m\ \theta_b\ \dot{\theta}_b\ \theta_s\ \dot{\theta}_s]^T$ as $\dot{\Theta} = \mathbf{A}\Theta + \mathbf{B}\mathbf{V}$, that is:

$$\underbrace{\begin{bmatrix}\dot{\theta}_m\\ \ddot{\theta}_m\\ \dot{\theta}_b\\ \ddot{\theta}_b\\ \dot{\theta}_s\\ \ddot{\theta}_s\end{bmatrix}}_{\dot{\Theta}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0\\ -\frac{k_s}{J_m} & -\frac{b_m}{J_m} & 0 & 0 & \frac{k_s}{J_m} & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ \frac{J_u k_s}{J} & 0 & \frac{J_s k_u}{J} & a_{44} & \frac{J_s k_s - J_u k_u}{J} & a_{46}\\ 0 & 0 & 0 & 0 & 0 & 1\\ \frac{(4J_b - J_u)k_s}{J} & 0 & \frac{J_b k_u}{J} & a_{46} & \frac{J_b k_u - 4J_b k_s + J_u k_s}{J} & a_{66}\end{bmatrix}}_{\mathbf{A}}\begin{bmatrix}\theta_m\\ \dot{\theta}_m\\ \theta_b\\ \dot{\theta}_b\\ \theta_s\\ \dot{\theta}_s\end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0\\ \frac{K_m}{J_m R_a} & 0\\ 0 & 0\\ 0 & \frac{K_h(J_u - 4J_s)}{R_b J}\\ 0 & 0\\ 0 & -\frac{K_h J_u}{R_b J}\end{bmatrix}}_{\mathbf{B}}\begin{bmatrix}V_m\\ V_b\end{bmatrix} \tag{10}$$

where the constants $a_{ij}$ in matrix $\mathbf{A}$ are

$$\begin{aligned} a_{44} &= \frac{J_u(2b_d + b_b) - J_s(4b_d + 4b_b - b_u)}{J}\\ a_{46} &= \frac{J_s(4b_d + b_u) - J_u(b_s + 2b_d)}{J}\\ a_{64} &= \frac{J_b(4b_d + b_u) - J_u(2b_d + b_b)}{J}\\ a_{66} &= \frac{J_u(b_s + 2b_d) + J_b(b_u - 4b_s - 4b_b)}{J}\end{aligned}$$

with $\mathbf{V} = \begin{bmatrix}V_m & V_b\end{bmatrix}^T$ being the input vector. When the actuator rotates, the output torque $\tau_{out}$ can be calculated based on the relative compression of both sides of the elastic element $\Delta\theta_s$, its stiffness constant $k_s$, and all dynamic losses as:

$$\tau_{out} = J_u\ddot{\theta}_u + 4b_d\dot{\theta}_u - 4b_d\dot{\theta}_b - 2k_s\Delta\theta_s \tag{11}$$

where $\Delta\theta_s = \theta_m - \theta_s$. By substituting $\theta_u$ for the differential law in (5), the governing equation for the application of the output torque

$$\tau_{out} = 2J_s\ddot{\theta}_s + 2b_s\dot{\theta}_s + 2b_d(\dot{\theta}_s - \dot{\theta}_b) + 2k_s(\theta_s - \theta_m) + \frac{J_u}{2}(\ddot{\theta}_s + \ddot{\theta}_b) + \frac{b_u}{2}(\dot{\theta}_s + \dot{\theta}_b) \tag{12}$$

By neglecting the minimal inertia in the spring-side and user-side differential bodies, the governing equation of the output torque becomes:

$$\tau_{out} = \left(\frac{b_u}{2} - 2b_d\right)\dot{\theta}_b + \left(\frac{b_u}{2} + 2b_d + 2b_s\right)\dot{\theta}_s - 2k_s\Delta\theta_s. \tag{13}$$

With the MISO state–space model known, the active disturbance rejection controller can be developed.

## 3. Multi-input active disturbance rejection torque controller

The objective of ADRC is to provide accurate torque outputs based strictly on measurements of the deflection of the spring. A reasonable estimate of the deflection angle can be extracted from the encoder readings on either side of the spring. Active disturbance rejection control (ADRC) will be used for this purpose, as ADRC is an error-based control method that can compensate for unmodelled disturbances, such as backlash and brake hysteresis, via a time-optimal solution to a reference trajectory designed for non-ideal systems. The output of the controller can be distributed to multiple inputs, which is the case for the actuator described earlier. Convergence of nonlinear ADRCs for multi-input systems is demonstrated in Guo and Zhao (2013).

### 3.1. Reference and transient output torque through spring deflection profile

In the context of torque control for the elastic actuator, the transient profile is an updated reference that is a function of the proportional error $e_1$ and time varying error $e_2$ for the *single output* system:

$$e_1(k) = \tau_{ref}(k) - \tau_{inf}(k) = \tau_{ref}(k) - k_s(\theta_m(k) - \theta_s(k)) \tag{14}$$

$$e_2(k) = e_2(k-1) + fhan(e_1(k), e_2(k-1), r_0, h_0) \tag{15}$$

where $\tau_{ref}$ is the reference torque, and $\tau_{inf} = k_s(\theta_m - \theta_s)$ is the inferred output torque measured from the deflection of the elastic element measured at sample $k$. These errors can be inserted into the *fhan* function from Han (2009) along with an acceleration coefficient $r_0$ and smoothing factor $h_0$ to produce a *desired transient profile* as shown in Fig. 2. The *fhan* function is defined as:

$$fhan = -r_0\left[\left(\frac{a}{h_0 r_0^2} - \text{sign}(a)\right)s_a + \text{sign}(a)\right] \tag{16}$$

This function creates distinct transient profiles with differing acceleration rates $r_0$ for the same reference. The output of *fhan* provides

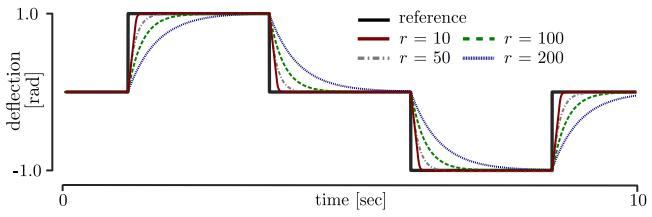



**Fig. 2.** Transient profiles generated with various acceleration rates $r = 10,\ 50,\ 100,\ 200$ for a reference deflection signal. The transient profile for the desired spring deflection provides an achievable alternative reference signal compared to the infinite derivative reference square profile.

a realistic alternative to transients in physical systems, as an input reference such as a Heaviside step function has an infinite derivative at the transient point which is impossible to recreate. The constants in (16) are (Han, 2009):

$$a_1^2 = h_0 r_0^2\left(h_0 r_0^2 + 8|e_1 + h_0 e_2|\right)$$
$$s_y = \text{sign}(e_1 + h_0 e_2 + h_0 r_0^2) - \text{sign}(e_1 + h_0 e_2 - h_0 r_0^2)/2$$
$$a_2 = h_0 e_2 + \text{sign}(e_1 + h_0 e_2)(a_1 - h_0 r_0^2)/2$$
$$a = (h_0 e_2 + e_1 + h_0 e_2 - a_2)s_y + a_2$$
$$s_a = \text{sign}(a + h_0 r_0^2)/2 - \text{sign}(a - h_0 r_0^2)/2$$

where $a$, $a_1$, $a_2$, $s_y$, and $s_a$ are intermediate variables.

The position and velocity reference commands for each of the subsystems of the multi-input actuator can be derived from the reference torque generated by the desired transient profile mentioned above. Once the desired profile is determined, (14) and (16) can be used to determine the controllable reference position and velocity states. The process is demonstrated for one of the subsystems (motor position and velocity) through the following discrete set of equations:

$$\tau_{ref}(k) = \tau_{ref}(k-1) + h\dot{\tau}_{ref}(k-1) \tag{17}$$
$$\dot{\tau}_{ref}(k) = \dot{\tau}_{ref}(k-1) + h * fhan(\tau_{inf}(k-1) - \tau_{ref}(k-1), \dot{\tau}_{ref}(k-1), r_0, h) \tag{18}$$

$$\theta_m(k) = \theta_m(k-1) + h\dot{\theta}_m(k-1) \tag{19}$$
$$\dot{\theta}_m(k) = \dot{\theta}_m(k-1) + h * fhan(\theta_m(k-1) - \frac{\tau_{ref}(k-1)}{k_s} - \theta_s(k-1), \dot{\theta}_m(k-1), r_0^1, h_0^1) \tag{20}$$

where $h$ is the sampling period and $r_0^1$ and $h_0^1$ are tuning parameters related to controller aggression and error levelling, respectively. The remaining subsystems can be computed in a similar manner to (19) and (20). This ensures each subsystem of the controller collectively tries to minimize the error $e_1(k)$ of the single output relative to the last sampled state of the actuator $(k-1)$.

### 3.2. Extended state observers

Consider the multi-input single-output (MISO) time varying system described in Section 2 with 6 state variables defined by the vector $\Theta_j \in \mathbb{R},\ j = 1,2,..,6$ as $\Theta = [x_1^1 \quad x_2^1 \quad x_1^2 \quad x_2^2 \quad x_1^3 \quad x_2^3]^T$ or $\Theta = [\theta_m \quad \dot{\theta}_m \quad \theta_s \quad \dot{\theta}_s \quad \theta_b \quad \dot{\theta}_b]^T$ as defined in (10). A block diagram of the controller is shown in Fig. 3. In this example, a system with three independent states $\theta_m, \theta_s$, and $\theta_b$ as well as their first time derivatives are measurable. Therefore, a total of three system equations for the ADRC can be used. The three subsystems $(i = 1,2,3)$ and their respective nonlinearities are:

$$\begin{cases} \dot{x}_1^i = f_1^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3, D^i(t)) \\ \dot{x}_2^i = f_2^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3, D^i(t)) + b^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3)u^i(t) \\ y^i = x_1^i \end{cases} \tag{21}$$

where $f_j, j = 1,2$, and $b$ are imperfect or nonlinear functions describing the subsystem and any external disturbances captured in $D(t)$, $u^i(t)$ is the control input of the subsystem, and $y^i(t)$ is the output, an angular displacement for the multi-input plant described in Section 2. The three local total disturbance terms can be estimated and combined by equating $\bar{x}_1^i = y^i$ and $\bar{x}_2^i = f_1^i(t, x_1^i, x_2^i, D^i(t))$, which provides the following description of the subsystem with a common disturbance term:

$$\begin{cases} \dot{\bar{x}}_1^i = \bar{x}_2^i \\ \dot{\bar{x}}_2^i = \frac{\partial \bar{x}_2^i}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_1^1}\frac{\partial x_1^1}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^1}\frac{\partial x_2^1}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_1^2}\frac{\partial x_1^2}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^2}\frac{\partial x_2^2}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_1^3}\frac{\partial x_1^3}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^3}\frac{\partial x_2^3}{\partial t} \\ \qquad + \frac{\partial \bar{x}_2^i}{\partial D^i}\frac{\partial D^i}{\partial t} \\ \qquad - \frac{\partial \bar{x}_2^i}{\partial x_2^i}\frac{\partial x_2^i}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^i} f_2^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3, D^i(t)) \\ \qquad + \frac{\partial \bar{x}_2^i}{\partial x_2^i} b^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3)u^i(t) \\ y^i = \bar{x}_1^i \end{cases} \tag{22}$$

A linear approximation $\bar{b}^i(t)$ for the nonlinear term $b^i$ allows the subsystem to be extended by a new state representing the sum of disturbances as $\bar{x}_3^i$. This disturbance and its first time derivative $\dot{\bar{x}}_3^i$ are defined by:

$$\begin{cases} \dot{\bar{x}}_2^i = \bar{x}_3^i + \bar{b}^i(t)u^i(t) \\ \dot{\bar{x}}_3^i = f_3^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3) \end{cases} \tag{23}$$

where the total disturbance of the subsystem can be combined to produce:

$$\begin{aligned} \bar{x}_3^i = {} & \frac{\partial \bar{x}_2^i}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_1^1}\frac{\partial x_1^1}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^1}\frac{\partial x_2^1}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_1^2}\frac{\partial x_1^2}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^2}\frac{\partial x_2^2}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_1^3}\frac{\partial x_1^3}{\partial t} \\ & + \frac{\partial \bar{x}_2^i}{\partial x_2^3}\frac{\partial x_2^3}{\partial t} \\ & + \frac{\partial \bar{x}_2^i}{\partial D^i}\frac{\partial D^i}{\partial t} - \frac{\partial \bar{x}_2^i}{\partial x_2^i}\frac{\partial x_2^i}{\partial t} + \frac{\partial \bar{x}_2^i}{\partial x_2^i} f_2^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3, D^i(t)) \\ & + \left(\frac{\partial \bar{x}_2^i}{\partial x_2^i} b^i(t, x_1^1, x_2^1, x_1^2, x_2^2, x_1^3, x_2^3) - \bar{b}^i(t)\right)u^i(t) \end{aligned} \tag{24}$$

The controller has three extended state observers (ESO) to determine the angular displacement and velocity of each of the subsystems. The ESO evaluates discrepancies in expected values and estimates disturbances present in each of the subsystems. The state extension of the ESO provides a means of evaluating nonlinearities around the spring deflection, magnetic hysteresis, static friction, and other unmodelled disturbances. Each ESO is defined as:

$$\begin{cases} \dot{\hat{x}}_1^i = \hat{x}_2^i - \beta_{01}^i g_1^i(\theta_m, \theta_b, \theta_s, \hat{y}^i(t)) \\ \dot{\hat{x}}_2^i = \hat{x}_3^i - \beta_{02}^i g_2^i(\theta_m, \theta_b, \theta_s, \hat{y}^i(t)) + \bar{b}^i(t)u \\ \dot{\hat{x}}_3^i = -\beta_{03}^i g_3^i(\theta_m, \theta_b, \theta_s, \hat{y}^i(t)) \\ \hat{y}^i = \hat{x}_1^i \end{cases} \tag{25}$$

where $\beta_{0j}^i\ j = 1,2,3$ are the observer gains for a dual integral plant state $i$ and $g^i(\theta_m, \theta_b, \theta_s, \hat{y}(t))$ is an error function for subsystem $i$. See Appendix A for the full derivation.

### 3.3. Control law

The issue surrounding multi-input systems lies in the derivation of the control law for each input. In systems employing a single extended state observer, the control law often is some combination of the *proposed control input* $u_p$ and a disturbance error correction. In *single input*

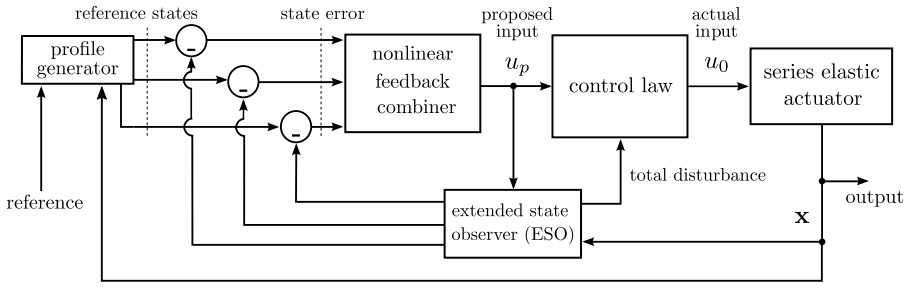



**Fig. 3.** Block diagram of a simplified version of the controller. The profile generator adjusts the input at unrealistic instantaneous reference transients to improve differential tracking error; the nonlinear feedback combiner aggregates the proportional and time-varying error in the states and proposes an input to the plant. The extended state observer provides a means of estimating and compensating unmodelled disturbances by creating a new state that encapsulates all disturbances in the system. This disturbance is fed into a control law that generates the actual control inputs for the MISO system.

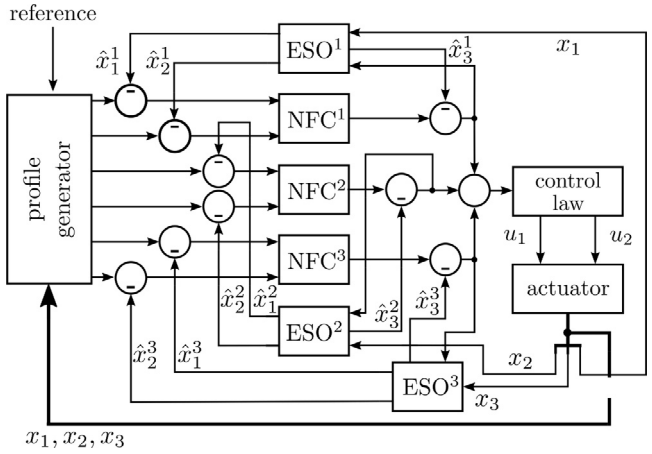



**Fig. 4.** Overall updated nonlinear ADRC scheme for a multi-input SEA. Three nonlinear feedback combiners (NFC) and three extended state observers (ESO) are utilized to adapt to perturbations in the spring deflection. The reference is first input into the transient profile generator, where an updated reference is determined. The estimated states from the ESOs are then combined into the NFCs and finally distributed to the multi-input system through the control law.

systems, the control law is similar to the following:

$$u(t) = -\left(\hat{x}_3(t) - u_p(t)\right)\left(\bar{b}\right)^{-1} \tag{26}$$

where $u_p(t)$ is from the nonlinear feedback combiner detailed below, and $k_1$ and $k_2$ are proportional and derivative control gains, respectively. Note that many valid linear or nonlinear controllers for $u_p$ exist. Potential single-input nonlinear operators are suggested by Han in Han (2009), where the proposed control input could be (see Fig. 3):

$$u_p = k_1\ fal(\tau_{ref} - \hat{\tau}_{inf}, \gamma_1, h) + k_2\ fal(\dot{\tau}_{ref} - \hat{\dot{\tau}}_{inf}, \gamma_2, h) \tag{27}$$

with $0 < \gamma_1 < 1 < \gamma_2$ being tuning parameters, and the nonlinear $fal$ function is

$$fal(e, \gamma, h) = \begin{cases} \frac{e}{h^{1-\gamma}}, & |e| \le h \\ |\ e\ |^{\gamma}\ \text{sign}(e), & |e| \ge h \end{cases} \tag{28}$$

designed to improve convergence time (Han, 2009). The goal of the nonlinear feedback combiner is to assist in converging at a faster rate than a PID controller. It is similar to producing time-varying PD gains. Since a static proportional and derivative gain can guarantee convergence for a reasonable range (Wang, Dodds, & Bailey, 1996), tuning the values of $\gamma_1$ and $\gamma_2$ can maintain this guarantee with the advantage of faster convergence times.

The primary method of controlling the torque in the multi-input single-output (MISO) SEA from Section 2 uses both controllable inputs, i.e., the commanded motor voltage $V_m$ and brake voltage $V_b$, in a distributive manner. The *actual control input* for the dual-input system is given as (see Fig. 4):

$$u_0 = u_1 + u_2 = f_1(k_s\Delta\theta_s)V_m^{max} + f_2(k_s\Delta\theta_s)V_b^{max} = \sum_{i=1}^{3} \frac{p_0^i(fhan(\varepsilon_1^i, \varepsilon_2^i, h_1^i, r_1^i) - \hat{x}_3^i)}{\bar{b}^i}, \tag{29}$$

$$u_1 = \sum_{i=1}^{3} \frac{p_1^i(fhan(\varepsilon_1^i, \varepsilon_2^i, h_1^i, r_1^i) - \hat{x}_3^i)}{\bar{b}^i}, \tag{30}$$

$$u_2 = \sum_{i=1}^{3} \frac{p_2^i(fhan(\varepsilon_1^i, \varepsilon_2^i, h_1^i, r_1^i) - \hat{x}_3^i)}{\bar{b}^i} \tag{31}$$

where $u_0$ represents the total control effort of the actuator, and $u_1$ and $u_2$ represent the input for the motor and brake, respectively. $p_0$ is a distributive gain consisting of the sum of all subsystem gains $p_j^i$, $j = 1, 2, 3$. These are tunable proportional parameters that map subsystem error to the causal inputs. A unity gain for a specific subsystem error function indicates that the cause of the error for the subsystem is directly addressed by the input in question. $\varepsilon_1$ and $\varepsilon_2$ are the observed proportional and time-varying error between the reference profiles generated in (19) and (20), and the observed states $\hat{x}_1$ and $\hat{x}_2$, respectively. These error functions are derived from the relationship between each subsystem and the single output $\tau_{out}$ given the current state of the actuator, i.e. the angular position and velocity of the remaining states. $r_1$ and $h_1$ are adjustable parameters unique to each subsystem. $\hat{x}_3^i$ is the total disturbance estimated by the observer from subsystem $i$. With the above control law the input distribution converges provided that

$$\forall k_s\Delta\theta_s < \tau_{ref} \exists [f_1(k_s\Delta\theta_s) > 0] \wedge [f_2(k_s\Delta\theta_s) > 0] \ni \Delta\dot{\theta}_s \ge 0. \tag{32}$$

and

$$\forall k_s\Delta\theta_s > \tau_{ref} \exists [f_1(k_s\Delta\theta_s) < 0 \wedge f_2(k_s\Delta\theta_s) > 0], \ \vee [f_1(k_s\Delta\theta_s) = 0 \wedge f_2(k_s\Delta\theta_s] = u_0/V_b) \ni \Delta\dot{\theta}_s \le 0$$

with $f_1 \in [-1, 1]$, $f_2 \in [0, 1]$ being linear distributing functions dependent on the current state of the spring deflection and the reference torque, and $p_0$ is a tuning parameter to produce meaningful distribution of the inputs. See Appendix B for the full derivation.

### 3.4. ADRC tunable parameters

Due to the large number of tunable parameters in the ADRC, it becomes difficult to evaluate the contribution of each parameter on the performance of the actuator. For each independent variable (inclusive of all time derivatives), there is one observer to estimate the states and one total disturbance term for the subsystem. The MISO systems in (29) have additional tunable gains equal to the number of inputs multiplied by the number of extended observers. Multi-input systems also have multiple transient profile generators. Each $fhan$ function defined in (16) has two controllable parameters, $r_0$ and $h_0$. Therefore, the number of tunable parameters from the transient profile generators is twice the number of extended state observers. All of the tunable ADRC

**Table 1**
Summary of all tunable parameters in an ADRC applied to series elastic actuator (SEA) control.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Motor subsystem | $\alpha_1^1$ | $\alpha_2^1$ | $\beta_{01}^1$ | $\beta_{02}^1$ | $\beta_{03}^1$ | $p_1^1$ | $p_2^1$ | $r_0^1$ | $h_0^1$ | $r_1^1$ | $h_1^1$ |
| Brake subsystem | $\alpha_1^2$ | $\alpha_2^2$ | $\beta_{01}^2$ | $\beta_{02}^2$ | $\beta_{03}^2$ | $p_1^2$ | $p_2^2$ | $r_0^2$ | $h_0^2$ | $r_1^2$ | $h_1^2$ |

parameters to be optimized in the following sections are summarized in Table 1.

## 4. Reference point dominance-based nondominated sorting genetic algorithm

The MISO ADRC described in the previous section has 22 tunable parameters. Finding approximate values for them becomes even more challenging with multiple control objectives, particularity when they contradict one another. Therefore, the controller tuning needs a multi-objective optimizer. As an example, consider two interdependent control objectives: the minimization of control effort and the minimization of rise time. Since rise time decreases with the control effort, if the latter is to be minimized the system would have an extended rise time. When the control objectives are in opposition, there might not be a single set of values for the ADRC parameters (called solution hereafter) to satisfy all of the control objectives. In fact, a set of optimal solutions create a *pareto front*, that is, a set of solutions that are not strictly inferior than (or are not dominated by) any other solution. A solution that performs better than another in at least one objective and not worse in any other objective is said to be dominant. Fig. 5 illustrates this phenomenon. The solutions connected with a dotted line represent the location of entirely nondominated solutions (pareto front). The dominated solutions have one or more solutions with a more optimal value for at least one of the objectives. This is represented in the figure by shaded regions, where the dominating solution lies on the bottom left corner of the regions and any other solution that is within this region is therefore dominated for the dual minimization objectives. In problems with multiple objectives, issues may arise around the diversity of solutions with complex pareto fronts (Ishibuchi, Setoguchi, Masuda, & Nojima, 2016; Li & Zhang, 2008).

A solution to automate the tuning of an ADRC should incorporate a multi-objective optimizer. As mentioned in the introduction, since the RPD-NSGA-II algorithm from Elarbi et al. (2018) is the most efficient of existing multi-objective solvers, it is selected to tackle the multi-objective problem presented in tuning the variables present in the ADRC presented earlier. In multi-objective optimization the goal is to obtain a set of potential solutions with satisfactory performance on all fronts. The chosen algorithm places emphasis on the diversity of the population in the evolutionary algorithm. The goal of the RPD-NSGA-II is to have a combination of convergence and diversity, which are not independent of one another. The stochastic nature of evolutionary algorithms helps improve diversity in the search for optimal solutions, the non-RPD-dominated sorting and selection of the multiple pareto fronts allows for convergence around the objectives.

### 4.1. Reference points and distance measures

The diversity guarantee of the RPD-NSGA-II algorithm is attributed to the reference points generated at the beginning of the process. The reference points are generated using a method proposed in Das and Dennis (1998), with a set of evenly distributed points on a normalized hyperplane as shown in Fig. 6(a). Once the fitness values for each objective are obtained, each range of fitness values are normalized by the maximum and minimum values, obtaining the solution set as marked by the green points in Fig. 6(b). The potential solutions are then associated with the nearest reference point vector from the ideal point (origin for minimization problems) along the normalized hyperplane. Consider the set of solutions in Fig. 6(b) that are nearing the plane

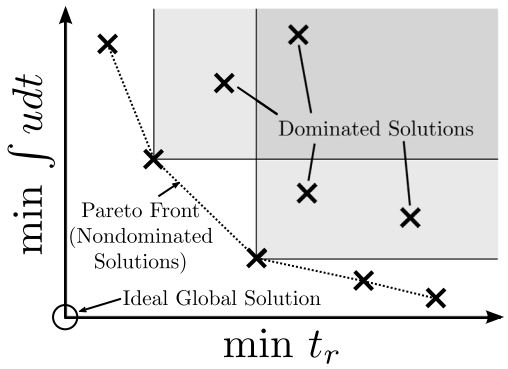



**Fig. 5.** Example of pareto front for control effort ($\int udt$) and rise time ($t_r$) minimization. The pareto front is a set of nondominated solutions with one or more solutions having an optimal value for at least one of the objectives.

$f_2 = 0$. A two-dimensional visualization is displayed in Fig. 6(c), where each of the solutions are attributed to a reference point. In this example, solution $a$ is associated with reference point $r_2$ and solutions $b$ and $c$ are associated with reference point $r_3$. Once all of the solutions have been associated with their respective closest reference point, each solution is assigned two distance measures, $d_1$ and $d_2$, used to aid in the non-RPD-dominated sorting process. Distance measure $d_1$ refers to the magnitude of the distance between the origin (for minimization problems) and the normal drawn from the reference vector to the potential solution, as in Fig. 6(c). This distance relates to convergence of the solutions, as a smaller magnitude of $d_1$ means better overall fitness of a solution. Distance measure $d_2$ refers to the magnitude of the normal as in Fig. 6(d). This distance is used to help encourage diversity in the selection process for the next generation, as reference points with solution crowding will begin to reduce emphasis on some of the solutions to favour diversity.

### 4.2. RP-dominance and non-RPD-dominated sorting

The two distance measurements are used to evaluate a solution's dominance over other solutions, thereby generating an alternative to determining pareto fronts apart from pareto-dominance. Let $\mathbf{z} = [z_1, z_2, \dots, z_m]$ be a vector containing the ADRC tunable parameters listed in Table 1. A potential solution $\mathbf{z_1}$ has a multi-objective fitness value

$$\mathbf{G}(\mathbf{z}_1) = [g_1(\mathbf{z}_1), g_2(\mathbf{z}_1), \dots, g_n(\mathbf{z}_1)] \tag{33}$$

is said to pareto dominate another solution, say $\mathbf{z}_2$, if

$$g_j(\mathbf{z_1}) \leq g_j(\mathbf{z_2})\ \forall j = 1, 2, \dots, n, \quad : \quad \mathbf{z}_2 \mid \mathbf{G}(\mathbf{z}_2) = [g_1(\mathbf{z}_2), g_2(\mathbf{z}_2), \dots, g_n(\mathbf{z}_2)] \tag{34}$$

and for at least one $j = 1, 2, \dots, n, \mid\ g_j(\mathbf{z_1}) < g_j(\mathbf{z_2})$. For the RPD-NSGA-II algorithm, dominance of a solution over another is taken a step further. Solution $\mathbf{z_1}$ is said to RP-dominate solution $\mathbf{z_2}$ if $\mathbf{z_1}$ pareto dominates $\mathbf{z_2}$ or if $\mathbf{z_1}$ and $\mathbf{z_2}$ are pareto equivalent and any of the following are true (Elarbi et al., 2018): **(1)** Both solutions are associated with the same reference point, but the value of $d_1$ for $\mathbf{z_1}$ is lower than the value of $d_1$ for $\mathbf{z_2}$; or **(2)** Both solutions are associated with different reference points, but the value of $d_1$ for $\mathbf{z_1}$ is lower than the value of $d_1$ for $\mathbf{z_2}$ and there are fewer solutions associated with the same reference point as $\mathbf{z_1}$ than that of $\mathbf{z_2}$. This operation is known as computing the reference point density.

Therefore, the entire population can be evaluated and placed into various dominating ranks. The ranks are an extension of pareto-dominance, with emphasis on diversity due to the second condition

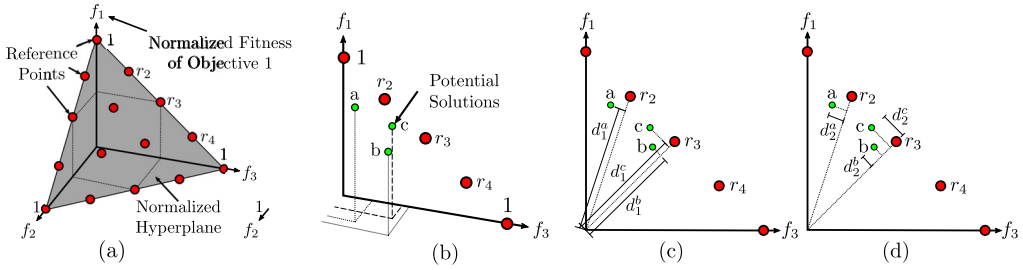



**Fig. 6.** Dominance-based non-dominated sorting genetic algorithm reference points and distance measures. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

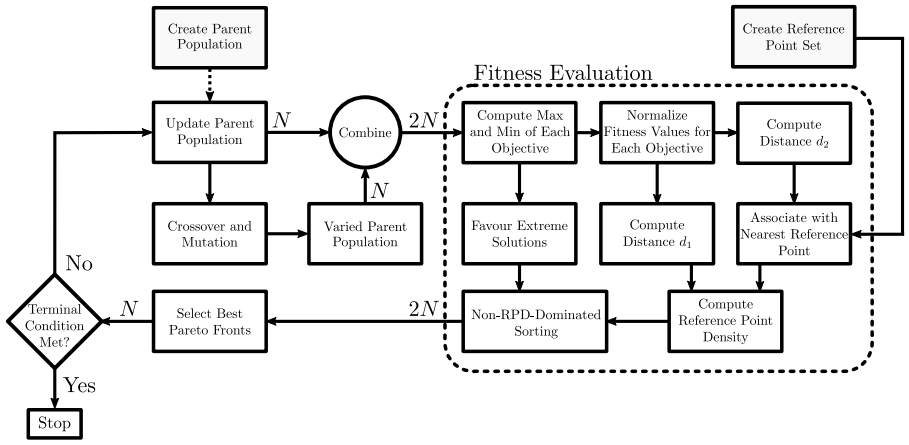



**Fig. 7.** Reference point dominance-based nondominated sorting genetic algorithm block diagram.

above. This methodology is referred to as non-RPD-dominated sorting (Elarbi et al., 2018). Once the entire population has been sorted, the top 50% of solutions amongst the best performing RP-dominated pareto fronts progress to the next generation of the optimization loop as the parent population. If the termination condition of the optimization process is not achieved, the newly recreated parent population is varied using standard stochastic genetic algorithm operators (crossover and mutation) and is re-evaluated for its fitness and distance measures. The total algorithm with all components is displayed in Fig. 7.

### 4.3. Performance objectives for SEA

One can now define the control objectives of the ADRC of the multi-input SEA as the following performance metrics. While all of these are well known to the control engineer, it is important to clarify how they relate to SEAs for human–machine interaction specifically.

**Tracking error** $E_t$: The strict adherence of a desired output state to a reference profile is the goal of most control systems and, therefore, this objective is biased in evaluating fitness in the optimizer. Tracking error refers to the integral of the error over all time. This is displayed as the sum of the shaded region in Fig. 8. Ideally, $E_t = 0$, which refers to perfect tracking between the reference $r(t)$ and the actual output $f(t)$. i.e.,

$$E_t = \int_0^t \mid r(t) - f(t) \mid dt \tag{35}$$

In the context of the SEA in human–machine interaction, minimal tracking error refers to the closeness of the desired torque to a reference torque. See Fig. 8.

**Control effort** $U_t$: Refers to the integral of the controller output $u_0(t)$ over all time:

$$U_t = \int_0^t \mid u_0(t) \mid dt \tag{36}$$

The controller effort relates to the actuator's efficiency, which constitutes a significant objective in portable human–robot interaction systems such as the SEA presented earlier.

**Percent overshoot** $P_o$: Refers to the percentage of the maximum output value that exceeds the reference. This is particularly useful when the output $f(t)$ is nearing rated hardware limits. $P_o$ is defined as:

$$P_o = \frac{\max f(t) - r(t)}{r(t)} \times 100\% \tag{37}$$

Minimizing the percent overshoot limits the maximum output torque in the SEA, which is particularly important in the context of force/torque control during human–machine interaction.

**Rise time** $t_r$: Rise time refers to the time it takes for the system to reach a value that is 95% of the reference value from the time of transience. This is useful for high speed switching applications. $t_r$ is defined by $t_r = \min(t) \mid f(t) \geq 0.95r(t)$.

**Settling time** ($t_s$): Refers to the time it takes for the system to permanently settle within ±5% of the reference value measured from the time of transience. The settling time $t_s$ is defined as $t_s = \max(t)$ where $\mid (f(t) - r(t)) \mid \geq 0.05r(t)$. Although the settling time is a good metric to determine stability in response to a transient state, as we shall see later on, minimizing the settling time in SEAs adds unwanted oscillations due to the presence of the elastic element.

**Maximum input** $u_{max}$: Refers to the maximum value of the input(s) of the device. This objective is useful when the source is limited, nearing saturation, or there are tight tolerances around the applied input to the plant. $u_{max}$ is defined as $u_{max} = \max(u(t))$. Maximum input is important for devices to be used in close proximity to humans. Note that the minimum objective for $u_{max}$ is the same as that for $U_t$, as both are trying to minimize the input.

**Steady-state error** $e_{ss}$: Steady-state error is the error between the reference and the settled output state when $t \to \infty$. This is particularly useful in high precision applications where there are tight tolerances on the final state of the output. The steady-state error $e_{ss}$ is defined

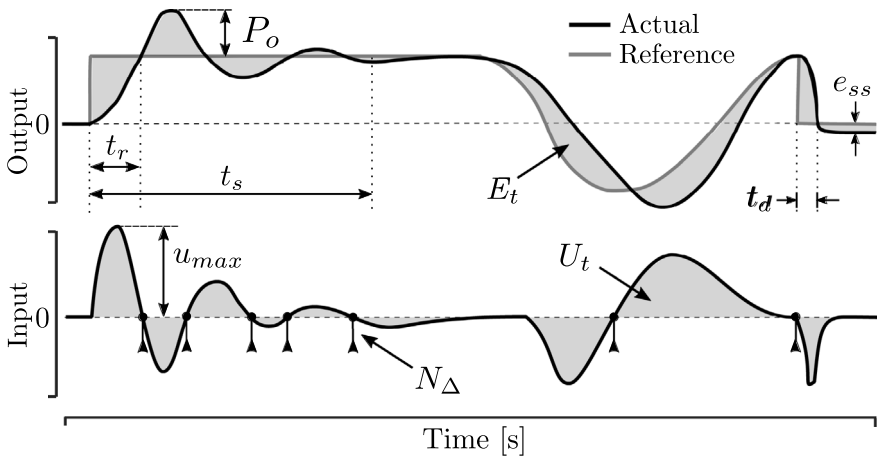



**Fig. 8.** The top figure demonstrates the response of a system $f(t)$ to a reference $r(t)$. The bottom figure represents the control output $u(t)$ for this device. The figure demonstrates common controller objectives such as tracking error $E_t$, control effort $U_t$, percent overshoot $P_o$, rise time $t_r$, settling time $t_s$, maximum input $U_{max}$, steady-state error $e_{ss}$, disengagement time $t_d$, and number of input direction changes $N_\Delta$.

as $e_{ss} = r(\infty) - f(\infty)$. Minimizing the steady-state error is important in SEAs where precise torque control is more desirable than the time it takes to reach the reference, allowing for smooth, well-defined motions.

**Disengagement time** $t_d$: Disengagement time is the time the SEA actuator takes to release the stored energy in the elastic element or reverse back to an initial state. It is measured from the time disengagement is initiated, $t_{oi}$, to the time when it is within 5% of the reference value at the time disengagement was initiated. This is important in applications regarding physical interaction with humans, as a source of collision mitigation. $t_d$ is defined as $t_d = \min(t) \mid f(t) \leq 0.05r(t_{oi})$. Quickly reducing the amount of energy in the actuator can ensure there is no holding torque and minimizes the amount of energy transferred to the output.

**Number of input direction changes** $N_\Delta$: Refers to the number of instances the control effort changes sign. High switching applications become difficult to physically implement and can damage sensitive plants. $N_\Delta$ is defined as:

$$N_\Delta = \sum \left| \frac{\text{sig}[u(t)] - \text{sig}[u(t-1)]}{2} \right| \tag{38}$$

This objective is inserted into the algorithm as a method to reduce the number of high-switching solutions from the population. High switching in a SEA is undesirable and may be an indication of an unstable behaviour.

Each of the above objectives are minimization functions, that is, the minimum value of each objective translates to better performance. A graphical representation of each of the above objective functions are displayed in Fig. 8.

## 5. Application to multi-input series elastic actuators

The ADRC and the optimizer described earlier can now be applied to control the SEA output torque. Fig. 1(b) shows the experimental setup. A similar reference profile to that shown in Fig. 8 is selected as the desired torque profile. It is assumed that the torque is proportional to the spring deflection, hence, the reference profile shown in Fig. 8 is the desired deflection of the SEA spring and is the input to the controller. This reference profile is selected to be challenging for the controller, as well as provide the necessary metrics to compute meaningful values for each control objective in Section 4.3.

The actuator's output shaft is connected to a load cell from which the output torque is measured. Due to this constraint and to ensure consistency in the experiments, $\theta_u = \dot{\theta}_u = 0 \; \forall t, \implies \theta_b = -\theta_s$, for all the scenarios considered later. The constrained condition of the actuator to measure the output torque using a load cell diminishes the need for an additional extended state observer. Thus, the controller has two extended state observers, one for controlling the contributions to the output deflection based on the motor's angular displacement and one for controlling the output deflection based on the angular displacement of the magnetic particle brake. Therefore, the controller has a total of 22 tunable parameters as listed in Table 1.

The state–space model of the actuator is implemented in the optimizer. For every generation, the entire merged population (parent population and varied population, see Fig. 7) is simulated and the fitness values for each objective is determined. The optimizer is chosen to have the crossover and mutation variation parameters set to 35, with a population of 300, over a total of 500 generations. To ensure the results provided meaningful control of the plant, a bias is placed on the tracking error. This provided much more desirable results, where the fitness values for increasing generations are displayed in Table 2 for a single objective at a time.

### 5.1. Experimental results

Once the optimization algorithm has run its course, the resulting pareto front set can be evaluated experimentally with the SEA. Two sets of experiments are reported: single and multiple objective optimization. The solution that provides the best result to minimize one or more control objectives at a time is selected and the gains determined from the optimizer are physically implemented.

### 5.2. Single control objective results

Table 2 summarizes the best results obtained in each objective and the experimental and simulation results are shown in Fig. 9. The top plot in each panel shows the desired deflection of the elastic element along with the simulated and measured deflections for the values listed in Table 2. The bottom panel presents the simulated and experimentally applied input to the actuator, i.e., the brake and motor voltages.

The ADRC gains are for the motor and brake found to minimize the tracking error are summarized in the first two lines of Table 3 and the experimental results obtained for these gains are displayed in Fig. 9a. From the experiments, one can discern that the actuator is able to maintain reasonable tracking of the spring deflection. The small perturbation around 12 s is due to backlash in the differential gear. The result is well matched with the simulated results and certainly performs favourable for reference tracking.

The optimized gains that minimize the control effort are listed in the third and fourth lines of Table 3 and the experimental results are shown in Fig. 9b. A net decrease in the motor voltage can be seen as

**Table 2**
Single objective performance over 500 generations.

| Generation | $E_t$ | $U_t$ | $P_o$ | $t_r$ | $t_s$ | $u_{max}$ | $e_{ss}$ | $t_d$ | $N_\Delta$ |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 0.9473 | 0.0259 | 31.91 | 0.158 | 0.564 | 0.0893 | 0.0061 | 0.004 | 5 |
| 20 | 0.8659 | 0.0251 | 35.75 | 0.154 | 0.558 | 0.0891 | 0.0023 | 0.004 | 5 |
| 30 | 0.8646 | 0.0245 | 26.09 | 0.154 | 0.558 | 0.0899 | 0.0008 | 0.010 | 5 |
| 40 | 0.8643 | 0.0236 | 25.82 | 0.154 | 0.558 | 0.0916 | 0.0012 | 0.006 | 5 |
| 50 | 0.8637 | 0.0224 | 22.43 | 0.154 | 0.556 | 0.0882 | 0.0003 | 0.076 | 5 |
| 100 | 0.8637 | 0.0219 | 15.59 | 0.162 | 0.556 | 0.0784 | 0.0002 | 0.142 | 5 |
| 150 | 0.8634 | 0.0220 | 17.89 | 0.124 | 0.556 | 0.0786 | 0.0003 | 0.114 | 5 |
| 200 | 0.8632 | 0.0236 | 15.31 | 0.160 | 0.556 | 0.0897 | 0.0003 | 0.142 | 5 |
| 250 | 0.8632 | 0.0249 | 25.53 | 0.124 | 0.556 | 0.1004 | 0.0008 | 0.008 | 3 |
| 300 | 0.8607 | 0.0270 | 25.70 | 0.134 | 0.558 | 0.1030 | 0.0018 | 0.004 | 3 |
| 350 | 0.8588 | 0.0288 | 10.97 | 0.124 | 0.882 | 0.1220 | 0.0036 | 0.004 | 0 |
| 400 | 0.8569 | 0.0256 | 1.260 | 0.124 | 1.222 | 0.1742 | 0.0117 | 0.006 | 0 |
| 450 | 0.8567 | 0.0238 | 4.580 | 0.126 | 0.556 | 0.0974 | 0.0007 | 0.022 | 0 |
| 500 | 0.8557 | 0.0225 | 0.037 | 0.126 | 0.554 | 0.0904 | 0.0010 | 0.046 | 0 |

**Table 3**
Optimized gains for individual and multiple control objectives.

| Gain | $\alpha_1^i$ | $\alpha_2^i$ | $\beta_{01}^i$ | $\beta_{02}^i$ | $\beta_{03}^i$ | $p_1^i$ | $p_2^i$ | $r_0^i$ | $h_0^i$ | $r_1^i$ | $h_1^i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Tracking error** | | | | | | | | | | |
| Motor ESO | 0.50 | 0.18 | 1.62 | 22.48 | 47.59 | 0.86 | 0.03 | 42.94 | 1.01 | 42.61 | 0.95 |
| Brake ESO | 0.47 | 0.14 | 1.56 | 17.84 | 41.18 | 0.40 | 0.97 | 38.19 | 0.98 | 32.36 | 0.95 |
| | **Controller effort** | | | | | | | | | | |
| Motor ESO | 0.46 | 0.18 | 1.75 | 26.16 | 40.39 | 0.85 | 0.14 | 39.07 | 0.87 | 44.16 | 0.98 |
| Brake ESO | 0.36 | 0.16 | 1.51 | 20.16 | 37.50 | 0.30 | 0.69 | 35.36 | 0.75 | 43.57 | 0.85 |
| | **Overshoot** | | | | | | | | | | |
| Motor ESO | 0.50 | 0.16 | 1.46 | 25.02 | 38.65 | 0.14 | 0.69 | 38.67 | 0.92 | 38.76 | 0.96 |
| Brake ESO | 0.46 | 0.14 | 1.34 | 22.95 | 35.45 | 0.14 | 0.64 | 35.47 | 0.85 | 35.55 | 0.89 |
| | **Rise time** (unstable system) | | | | | | | | | | |
| Motor ESO | 0.19 | 0.49 | 1.38 | 24.59 | 81.02 | 0.77 | 0.33 | 32.03 | 0.08 | 43.23 | 0.28 |
| Brake ESO | 0.15 | 0.48 | 1.24 | 22.99 | 64.82 | 0.17 | 0.63 | 30.57 | 0.08 | 41.65 | 0.27 |
| | **Settling time** | | | | | | | | | | |
| Motor ESO | 0.48 | 0.19 | 1.67 | 25.24 | 46.31 | 0.74 | 0.23 | 42.40 | 1.02 | 41.47 | 0.96 |
| Brake ESO | 0.41 | 0.18 | 1.31 | 21.31 | 42.25 | 0.21 | 0.84 | 41.60 | 0.96 | 40.19 | 0.92 |
| | **Maximum input** | | | | | | | | | | |
| Motor ESO | 0.46 | 0.18 | 1.75 | 26.16 | 40.39 | 0.85 | 0.14 | 39.07 | 0.87 | 44.16 | 0.98 |
| Brake ESO | 0.36 | 0.16 | 1.51 | 20.16 | 37.50 | 0.30 | 0.69 | 35.36 | 0.75 | 43.57 | 0.85 |
| | **Steady-state error** | | | | | | | | | | |
| Motor ESO | 0.48 | 0.18 | 1.68 | 24.84 | 40.25 | 0.96 | 0.37 | 40.62 | 0.79 | 45.54 | 0.96 |
| Brake ESO | 0.37 | 0.15 | 1.65 | 23.38 | 36.70 | 0.33 | 0.99 | 40.18 | 0.75 | 38.15 | 0.78 |
| | **Time to disengagement** | | | | | | | | | | |
| Motor ESO | 0.09 | 0.37 | 1.51 | 47.34 | 49.20 | 0.80 | 0.18 | 29.82 | 1.65 | 71.74 | 0.61 |
| Brake ESO | 0.08 | 0.34 | 1.23 | 45.10 | 40.75 | 0.07 | 0.85 | 25.16 | 1.33 | 70.29 | 0.47 |
| | **Number of input crossings** | | | | | | | | | | |
| Motor ESO | 0.19 | 0.49 | 1.38 | 24.59 | 81.02 | 0.77 | 0.33 | 32.03 | 0.08 | 43.23 | 0.28 |
| Brake ESO | 0.42 | 0.17 | 1.57 | 21.82 | 36.97 | 0.33 | 0.68 | 37.19 | 0.81 | 35.88 | 0.84 |
| | **Multiobjective: tracking error, control effort, and overshoot** | | | | | | | | | | |
| Motor ESO | 0.30 | 0.02 | 1.59 | 25.98 | 40.23 | 0.73 | 0.21 | 38.91 | 0.71 | 44.00 | 0.82 |
| Brake ESO | 0.36 | 0.16 | 1.48 | 22.32 | 38.30 | 0.12 | 0.81 | 36.83 | 0.69 | 42.01 | 0.93 |

compared to Fig. 9a. One can also note the absence of any overshoot, which is another indication that the control objective has been attained.

The results for minimal overshoot are shown in Fig. 9c. One can again observe the lack of overshoot. Further, the control effort resembles again that of Fig. 9a. The results for the settling time, maximum input, and state error in both the simulations and the experiments for the gains highlighted in Table 3 are displayed in Figs. 9d, 9e, and 9f, respectively. From the results it can be seen that each control objective is achieved. The results of the controller minimizing time to disengagement using the gains in Table 3 in both the simulations and experiments are displayed in Fig. 9g. In this objective the fitness value is a function of the number of input direction changes, therefore, naturally unbiasing high switching solutions.

The minimization of rise time resulted in an unstable controller. The final population of the 500th generation contained members with unstable controllers, however, this instability does ensure the output rises as quickly to the reference as possible. This is a good example on the importance of defining objectives and biasing the outcome based on the more important objectives. If there were no bias on any objective, the unstable controller solution would become selected for variation into future generations. For the most part, this can have adverse effects on the progression of the controller gains, as it contradicts most of

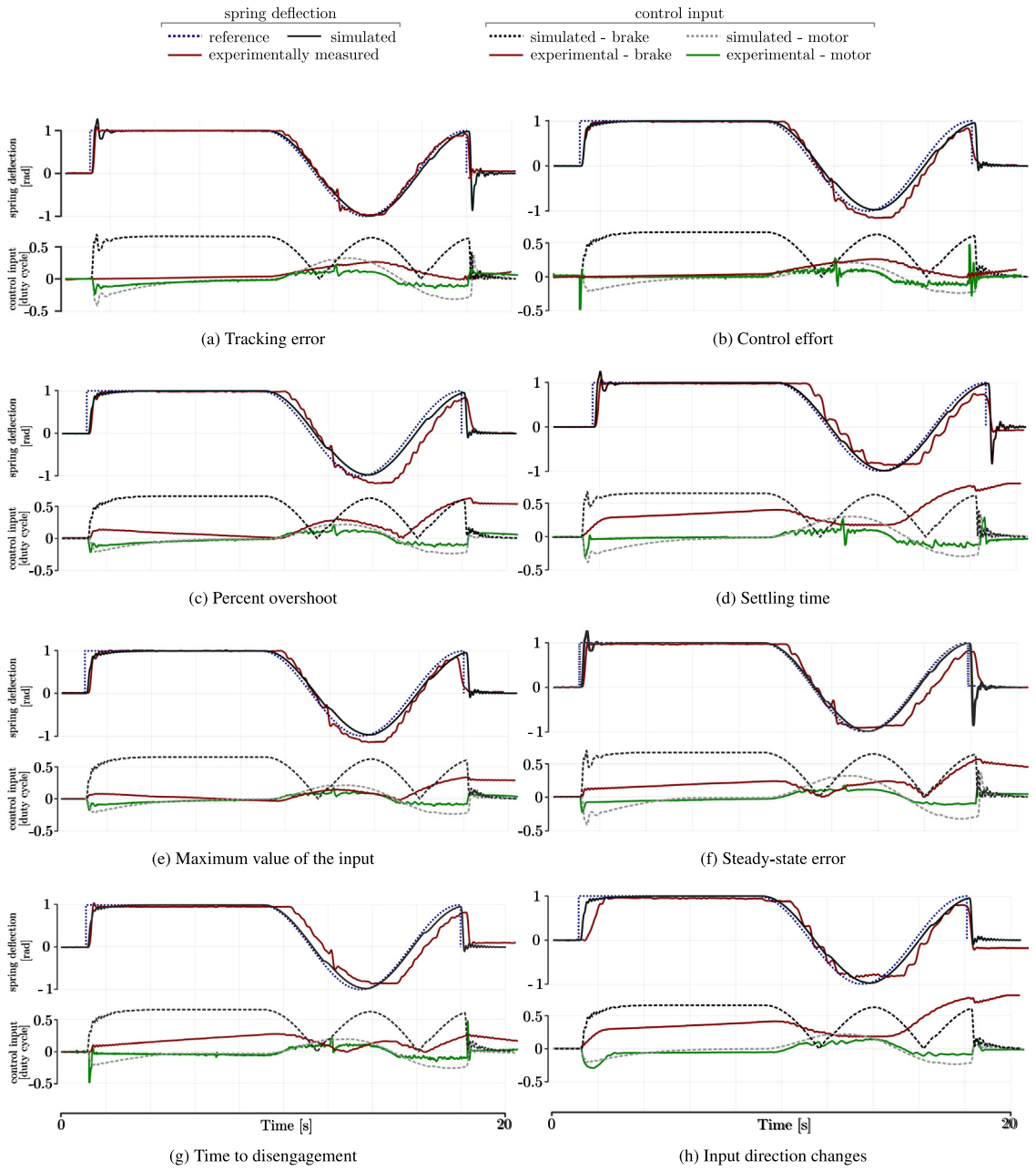



**Fig. 9.** Single objective experimental results. The best result obtained from using the controller gains determined by the optimizer biased towards minimizing each control objective.

the other objectives. For obvious reasons, the controller was not implemented experimentally and the simulation results are not included in Fig. 9.

### 5.3. Multi-objective control results

The optimizer has demonstrated validity in optimizing various single objective cases, however, one has the option to choose a solution that best represents their unique application from the population set at the end of the optimization. Consider 5 members of the resulting population shown in Table 4, and their respective fitness values when the tracking error, control effort, and overshoot, are set as concurrent control objectives.

If the device was to be used in the context of human–machine interaction, specifically robot-assisted rehabilitation, there may be a number of deterrents when selecting the appropriate control strategy. For example, there may be a very specific torque goal in mind and to ensure that the patient does not experience overexertion and perhaps the device is destined to be mobile and, therefore, battery operated. In this case, the most significant objectives to optimize are the tracking error, control effort, and percent overshoot. If these were the specifications for the controller design, the three candidate solutions could be Members 1, 2, 3, and 5 of the population from Table 4, as each of them have reasonable values for minimizing the three objectives in question. Member 4 may be discarded, as it is dominated by every other solution with respect to the significant objectives. Furthermore, Members 1 and 5 have a percent overshoot that could be considered unreasonably high for the design specifications and, therefore, could be discarded as well. The remaining members, 2 and 3, have relatively similar values for the tracking error $E_t$ and the percent overshoot $P_o$, but vary significantly in control effort $U_t$. The designer may also choose

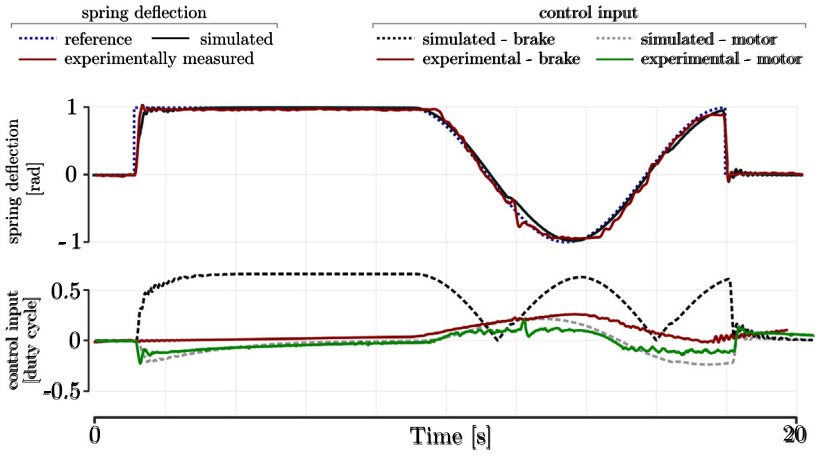



**Fig. 10.** Experimental results for the multi-objective optimization, as selected by the designer.

**Table 4**
Multi-objective fitness values for five members.

| Member | $E_t$ | $U_t$ | $P_o$ | $t_r$ | $t_s$ | $u_{max}$ | $e_{ss}$ | $t_d$ | $N_\Delta$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **0.8678** | 0.0281 | 30.869 | 0.1740 | 0.8920 | 0.1045 | 0.0076 | 0.1480 | **5** |
| 2 | 0.9615 | 0.1370 | **0.581** | 0.1680 | 1.2220 | 0.2141 | 0.0659 | 0.1440 | 7 |
| 3 | 1.1783 | **0.0259** | 1.237 | 0.7940 | 1.0140 | **0.0942** | 0.0092 | 0.1860 | **5** |
| 4 | 8.3227 | 0.6353 | 122.80 | **0.1500** | 6.5020 | 0.4812 | 236.47 | **0.1320** | 65 |
| 5 | 0.9098 | 0.0307 | 34.021 | 0.1700 | **0.8820** | 0.1204 | **0.0068** | 0.1460 | **5** |

to select the gains optimized from Member 3 as it has the lowest control effort, compromising slightly on tracking error and percent overshoot compared to Member 2.

The results of Member 3 with optimized gains from Table 4 from the final population set is demonstrated in Fig. 10. This example demonstrates the importance of multi-objective optimization in the process of selecting controller gains for specific applications, where the designer can view the trade-offs between various solutions. The ability to gauge the overall performance of a controller provides a means of tailoring the controller based on the specifications of the application. From the experimental results, it is clear that the multi-objective optimizer can combine the features of several single objective problems described in the previous section.

## 6. Conclusions

The most difficult portion of active disturbance rejection control is the tuning of the system parameters. Presented in this paper is the implementation of the RPD-NSGA-II from Elarbi et al. (2018) to optimize the parameters required for ADRC on a multi-input SEA. 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 designer is able to choose gain values based on the objectives most suitable for their applications. Comparing the results of this paper with other optimization algorithms is difficult since this is the only implementation of a generic algorithm to a dual input SEA. However, for a comparison between the performance of this optimizer and other decomposition-based multi objective genetic algorithms, the reader is refereed to Elarbi et al. (2018), and a comparison between a PID controller and an ADRC applied to the same SEA is available in DeBoon, Nokleby, and Rossa (2020).

In order to choose gains that are favourable to multiple objectives, the designer could evaluate the set of fitness values for each member of the resulting population and determine how they want to bias their controller. The relative trade-offs between the objectives becomes apparent and, therefore, the designer can select the set of gains that is best suited for their applications. There are a wide number of applications for an ADRC optimization method. Since active disturbance rejection control is a favourable alternative to PID, the controller can be used in a multitude of plants ranging from robotic actuators used in medical devices to autonomous vehicles and industrial automation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. Expanded state observers for the multi-input series elastic actuator

The three extended state observers can be determined from the equalities: $\bar{x}_1^1 = \theta_m$, $\bar{x}_1^2 = \theta_b$, and $\bar{x}_1^3 = \theta_s$. The systems of equations for the subsystems are structured as follows. For the motor subsystem:

$$\begin{cases} \dot{\bar{x}}_1^1 = \bar{x}_2^1 \\ \dot{\bar{x}}_2^1 = -\frac{k_s}{J_m} x_1^1 - \frac{b_m}{J_m} x_2^1 + \frac{k_s}{J_m} x_1^3 + D^1(t) + f_m(t,\Theta) V_m \\ y^1 = \bar{x}_1^1 \end{cases} \tag{A.1}$$

For the brake subsystem:

$$\begin{cases} \dot{\bar{x}}_1^2 = \bar{x}_2^2 \\ \dot{\bar{x}}_2^2 = \frac{J_u k_s}{J} x_1^1 + \frac{J_s k_u}{J} x_1^2 + \frac{J_u(2b_d+b_b)-J_s(4b_d+4b_b-b_u)}{J} x_2^2 + \frac{J_s k_u - J_u k_s}{J} x_1^3 \\ + \frac{J_s(4b_d+b_u)-J_u(b_s+2b_d)}{J} x_2^3 + D^2(t) + f_{b1}(t,\Theta) V_b \\ y^2 = \bar{x}_1^2 \end{cases} \tag{A.2}$$

For the spring subsystem:

$$\begin{cases} \dot{\bar{x}}_1^3 = \bar{x}_2^3 \\ \dot{\bar{x}}_2^3 = \frac{(4J_b - J_u)k_s}{J} x_1^1 + \frac{J_b k_u}{J} x_1^2 + \frac{J_b(4b_d + b_u) - J_u(2b_d + b_b)}{J} x_2^2 + \frac{J_b k_u - 4J_b k_s + J_u k_s}{J} x_1^3 \\ + \frac{J_u(b_s + 2b_d) + J_b(b_u - 4b_s - 4b_b)}{J} x_2^3 + D^3(t) + f_{b2}(t, \Theta) V_b \\ y^3 = \bar{x}_1^3 \end{cases} \tag{A.3}$$

where $J = 4J_s J_b - J_u J_s - J_u J_b$. These equations can be converted to three extended state variables, each with their own total disturbance terms $\bar{x}_3^i$. For the motor subsystem:

$$\begin{cases} \dot{\bar{x}}_1^1 = \bar{x}_2^1 \\ \dot{\bar{x}}_2^1 = \bar{x}_3^1 + \frac{K_m K_v}{J_m R_a} V_m \\ y^1 = \bar{x}_1^1, \\ \bar{x}_3^1 = -\frac{k_s}{J_m} x_1^1 - \frac{b_m}{J_m} x_2^1 + \frac{k_s}{J_m} x_1^3 + D^1(t) + \left( f_m(t, \Theta) - \frac{K_m K_v}{J_m R_a} \right) V_m \end{cases} \tag{A.4}$$

The linear approximation of the input function in this case is $f_m(t, \Theta) \cong \frac{K_m K_v}{J_m R_a}$. Therefore, $\bar{b}^1 = \frac{K_m K_v}{J_m R_a}$. Similarly for the brake subsystem:

$$\begin{cases} \dot{\bar{x}}_1^2 = \bar{x}_2^2 \\ \dot{\bar{x}}_2^2 = \bar{x}_3^2 + \frac{K_h(J_u - 4J_s)}{R_b J} V_b \\ y^2 = \bar{x}_1^2, \\ \bar{x}_3^1 = \frac{J_u k_s}{J} x_1^1 + \frac{J_s k_u}{J} x_1^2 + \frac{J_u(2b_d + b_b) - J_s(4b_d + 4b_b - b_u)}{J} x_2^2 + \frac{J_s k_u - J_u k_s}{J} x_1^3 \\ \quad + \frac{J_s(4b_d + b_u) - J_u(b_s + 2b_d)}{J} x_2^3 + D^2(t) + \left( f_{b1}(t, \Theta) - \frac{K_h(J_u - 4J_s)}{R_b J} \right) V_b \end{cases} \tag{A.5}$$

The linear approximation of the input function in this case is $f_{b1}(t, \Theta) \cong \frac{K_h(J_u - 4J_s)}{R_b J}$. Therefore, $\bar{b}^2 = K_h(J_u - 4J_s)/(R_b J)$. Finally for the spring subsystem:

$$\begin{cases} \dot{\bar{x}}_1^3 = \bar{x}_2^3 \\ \dot{\bar{x}}_2^3 = \bar{x}_3^3 - \frac{K_h J_u}{R_b J} V_b \\ y^3 = \bar{x}_1^3, \\ \bar{x}_3^1 = \frac{(4J_b - J_u)k_s}{J} x_1^1 + \frac{J_b k_u}{J} x_1^2 + \frac{J_b(4b_d + b_u) - J_u(2b_d + b_b)}{J} x_2^2 + \frac{J_b k_u - 4J_b k_s + J_u k_s}{J} x_1^3 \\ + \frac{J_u(b_s + 2b_d) + J_b(b_u - 4b_s - 4b_b)}{J} x_2^3 + D^3(t) + \left( f_{b2}(t, \Theta) + \frac{K_h J_u}{R_b J} \right) V_b \end{cases} \tag{A.6}$$

The linear approximation of the input function in this case is $f_{b2}(t, \Theta) \cong -\frac{K_h J_u}{R_b J}$. Therefore, $\bar{b}^3 = -(K_h J_u)/(R_b J)$. The three extended state observers for the actuator are defined by:

Motor subsystem:

$$\begin{cases} \dot{\hat{x}}_1^1 = \hat{x}_2^1 - \beta_{01}^1 g_1^1(\theta_m, \theta_b, \theta_s, \hat{y}^1(t)) \\ \dot{\hat{x}}_2^1 = \hat{x}_3^1 - \beta_{02}^1 g_2^1(\theta_m, \theta_b, \theta_s, \hat{y}^1(t)) + \bar{b}^1(t) u^1 \\ \dot{\hat{x}}_3^1 = -\beta_{03}^1 g_3^1(\theta_m, \theta_b, \theta_s, \hat{y}^1(t)) \\ \hat{y}^1 = \hat{x}_1^1 \end{cases} \tag{A.7}$$

where $\beta_{0j}^1$, $j = 1, 2, 3$ are observer proportional coefficients selected by the designer. $g_1^j$, $j = 1, 2, 3$ are observer error functions. The error function suggested from Han (2009) provide nonlinear observer error functions as:

$$g_1^1(\theta_m, \theta_b, \theta_s, \hat{y}^1(t)) = (\hat{x}_1^1 - x_1^1) = (\hat{\theta}_m - \theta_m)$$
$$g_2^1(\theta_m, \theta_b, \theta_s, \hat{y}^1(t)) = fal(\hat{x}_1^1 - x_1^1, \alpha_1^1, h) = fal(\hat{x}_1^1 - x_1^1, 0.5, h)$$
$$g_3^1(\theta_m, \theta_b, \theta_s, \hat{y}^1(t)) = fal(\hat{x}_1^1 - x_1^1, \alpha_2^1, h) = fal(\hat{x}_1^1 - x_1^1, 0.25, h)$$

where $\theta_m$ is determined by the profile generator in Eq. (19) and the $fal$ function is defined in Eq. (28). Similarly the brake and spring subsystems, respectively, are:

$$\begin{cases} \dot{\hat{x}}_1^2 = \hat{x}_2^2 - \beta_{01}^2 g_1^2(\theta_m, \theta_b, \theta_s, \hat{y}^2(t)) \\ \dot{\hat{x}}_2^2 = \hat{x}_3^2 - \beta_{02}^2 g_2^2(\theta_m, \theta_b, \theta_s, \hat{y}^2(t)) + \bar{b}^2(t) u^2 \\ \dot{\hat{x}}_3^2 = -\beta_{03}^2 g_3^2(\theta_m, \theta_b, \theta_s, \hat{y}^2(t)) \\ \hat{y}^2 = \hat{x}_1^2 \\ g_1^2(\theta_m, \theta_b, \theta_s, \hat{y}^2(t)) = (\hat{x}_1^2 - x_1^2) = (\hat{\theta}_b - \theta_b) \\ g_2^2(\theta_m, \theta_b, \theta_s, \hat{y}^2(t)) = fal(\hat{x}_1^2 - x_1^2, \alpha_1^2, h) = fal(\hat{x}_1^2 - x_1^2, 0.5, h) \\ g_3^2(\theta_m, \theta_b, \theta_s, \hat{y}^2(t)) = fal(\hat{x}_1^2 - x_1^2, \alpha_2^2, h) = fal(\hat{x}_1^2 - x_1^2, 0.25, h) \end{cases}$$

$$\begin{cases} \dot{\hat{x}}_1^3 = \hat{x}_2^3 - \beta_{01}^3 g_1^3(\theta_m, \theta_b, \theta_s, \hat{y}^3(t)) \\ \dot{\hat{x}}_2^3 = \hat{x}_3^3 - \beta_{02}^3 g_2^3(\theta_m, \theta_b, \theta_s, \hat{y}^3(t)) + \bar{b}^3(t) u^3 \\ \dot{\hat{x}}_3^3 = -\beta_{03}^3 g_3^3(\theta_m, \theta_b, \theta_s, \hat{y}^3(t)) \\ \hat{y}^3 = \hat{x}_1^3 \\ g_1^3(\theta_m, \theta_b, \theta_s, \hat{y}^3(t)) = (\hat{x}_1^3 - x_1^3) = (\hat{\theta}_s - \theta_s) \\ g_2^3(\theta_m, \theta_b, \theta_s, \hat{y}^3(t)) = fal(\hat{x}_1^3 - x_1^3, \alpha_1^3, h) = fal(\hat{x}_1^3 - x_1^3, 0.5, h) \\ g_3^3(\theta_m, \theta_b, \theta_s, \hat{y}^3(t)) = fal(\hat{x}_1^3 - x_1^3, \alpha_2^3, h) = fal(\hat{x}_1^3 - x_1^3, 0.25, h) \end{cases}$$

## Appendix B. Expanded control law

The overall control law for the system can be described as follows:

$$u_q(t) = -\left( p_q^1 \frac{\hat{x}_3^1(t) - u_p^1}{\bar{b}^1} + p_q^2 \frac{\hat{x}_3^2(t) - u_p^2}{\bar{b}^2} + p_q^3 \frac{\hat{x}_3^3(t) - u_p^3}{\bar{b}^3} \right) \tag{B.1}$$

where $u_q(t)$ is the overall control input to the plant, $q = 1$ relates to the motor voltage and $q = 2$ relates to the brake input voltage, $p_q^i$, $i = 1, 2, 3$ are proportional input contribution gains, and $u_p^i$, $i = 1, 2, 3$ are the proposed input contribution for each of the three nonlinear feedback combiners. Let: $\varepsilon_m = \hat{\theta}_m - \theta_m$, $\varepsilon_b = \hat{\theta}_b - \theta_b$, $\varepsilon_s = \hat{\theta}_s - \theta_s$. The control law for the MISO system can be defined as follows:

$$u_q(t) = p_q^1 \frac{fhan(\varepsilon_m, \dot{\varepsilon}_m, h_1^1, r_1^1) - \hat{x}_3^1(t)}{\bar{b}^1} +$$
$$p_q^2 \frac{fhan(\varepsilon_b, \dot{\varepsilon}_b, h_1^2, r_1^2) - \hat{x}_3^2(t)}{\bar{b}^2} + p_q^3 \frac{fhan(\varepsilon_s, \dot{\varepsilon}_s, h_1^3, r_1^3) - \hat{x}_3^3(t)}{\bar{b}^3}$$

An alternative control law to the one proposed in (29) is described by the following nonlinear PD feedback combiner provided that $0 < \gamma_1^i < 1 < \gamma_2^i$ and $h$ is the sampling period:

$$u_q(t) = p_q^1 \frac{(k_1^1\ fal(\varepsilon_m, \gamma_1^1, h) + k_2^1\ fal(\dot{\varepsilon}_m, \gamma_2^1, h)) - \hat{x}_3^1(t)}{\bar{b}^1} +$$
$$p_q^2 \frac{(k_1^2\ fal(\varepsilon_b, \gamma_1^2, h) + k_2^2\ fal(\dot{\varepsilon}_b, \gamma_2^2, h)) - \hat{x}_3^2(t)}{\bar{b}^2}$$
$$+ p_q^3 \frac{(k_1^3\ fal(\varepsilon_s, \gamma_1^3, h) + k_2^3\ fal(\dot{\varepsilon}_s, \gamma_2^3, h)) - \hat{x}_3^3(t)}{\bar{b}^3}$$

## References


Ahi, B., & Nobakhti, A. (2017). Hardware implementation of an adrc controller on a gimbal mechanism. *IEEE Transactions on Control Systems Technology*.

Ciro, G. C., Dugardin, F., Yalaoui, F., & Kelly, R. (2016). A nsga-ii and nsga-iii comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine*, *49*(12), 1272–1277.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, *8*(3), 631–657.

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints.. *IEEE Transactions on Evolutionary Computation*, *18*(4), 577–601.

Deb, K., & Saxena, D. (2006). 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)* (pp. 3352–3360).

DeBoon, B., Nokleby, S., La Delfa, N., & Rossa, C. (2019). Differentially-clutched series elastic actuator for robot-aided musculoskeletal rehabilitation. In *2019 international conference on robotics and automation (ICRA)* (pp. 1507–1513). IEEE.

DeBoon, B., Nokleby, S., & Rossa, C. (2020). Backlash-compensated active disturbance rejection control of nonlinear multi-input series elastic actuators. In *International conference on robotics and automation (ICRA)* (pp. 6183–6189). IEEE.

Du, C., Yin, Z., Zhang, Y., Liu, J., Sun, X., & Zhong, Y. (2018). Research on active disturbance rejection control with parameter autotune mechanism for induction motors based on adaptive particle swarm optimization algorithm with dynamic inertia weight. *IEEE Transactions on Power Electronics*, *34*(3), 2841–2855.

Elarbi, M., Bechikh, S., Gupta, A., Said, L. B., & Ong, Y.-S. (2018). A new decomposition-based nsga-ii for many-objective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(7), 1191–1210.

Geng, H., Yang, H., Zhang, Y., & Chen, H. (2010). Auto-disturbances-rejection controller design and it's parameter optimization for aircraft longitudinal attitude. *Journal of System Simulation*, *22*(1), 89–91.

Guo, B.-Z., & Zhao, Z.-L. (2013). On convergence of the nonlinear active disturbance rejection control for mimo systems. *SIAM Journal on Control and Optimization*, *51*(2), 1727–1757.

Han, J. (2009). From pid to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, *56*(3), 900–906.

Hernandez Mejia, J. A., Schütze, O., Cuate, O., Lara, A., & Deb, K. (2017). Rds-nsga-ii: a memetic algorithm for reference point based multi-objective optimization. *Engineering Optimization*, *49*(5), 828–845.

Hou, G., Wang, M., Gong, L., & Zhang, J. (2018). 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)* (pp. 497–501). IEEE.

Hu, K., Zhang, X.-f., & Liu, C.-b. (2013). Unmanned underwater vehicle depth adrc based on genetic algorithm near surface. *Acta Armamentarii*, *34*(2), 217–222.

Ishibuchi, H., Setoguchi, Y., Masuda, H., & Nojima, Y. (2016). Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, *21*(2), 169–190.

Kalyanmoy, D., et al. (2001). *Multi objective optimization using evolutionary algorithms*. John Wiley and Sons.

Li, H., & Zhang, Q. (2008). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, *13*(2), 284–302.

Li, F., Zhang, Z., Armaou, A., Xue, Y., Zhou, S., & Zhou, Y. (2018). Study on adrc parameter optimization using cpso for clamping force control system. *Mathematical Problems in Engineering*, *2018*.

Madoński, R., Piosik, A., & Herman, P. (2013). High-gain disturbance observer tuning seen as a multicriteria optimization problem. In *21st Mediterranean conference on control and automation* (pp. 1411–1416). IEEE.

Wang, J., Dodds, S., & Bailey, W. (1996). Guaranteed rates of convergence of a class of PD controllers for trajectory tracking problems of robotic manipulators with dynamic uncertainties. *IEE Proceedings D (Control Theory and Applications)*, *143*(2), 186–190.

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

Wang, Z., Zu, R., Duan, D., & Li, J. (2019). Tuning of adrc for qtr in transition process based on nbpo hybrid algorithm. *IEEE Access*, *7*, 177219–177240.

Wu, G., Sun, L., & Lee, K. Y. (2017). Disturbance rejection control of a fuel cell power plant in a grid-connected system. *Control Engineering Practice*, *60*, 183–192.

Xing, H.-L., Jeon, J.-H., Park, K., & Oh, I.-K. (2013). Active disturbance rejection control for precise position tracking of ionic polymer–metal composite actuators. *IEEE/ASME Transactions on Mechatronics*, *18*(1), 86–95.

Yin, Z., Du, C., Liu, J., Sun, X., & Zhong, Y. (2018). Research on autodisturbance-rejection control of induction motors based on an ant colony optimization algorithm. *IEEE Transactions on Industrial Electronics*, *65*(4), 3077–3094.

Yuan, Y., Xu, H., & Wang, B. (2014). An improved nsga-iii procedure for evolutionary many-objective optimization. In *Proceedings of the 2014 annual conference on genetic and evolutionary computation* (pp. 661–668). ACM.

Zhang, Y., Fan, C., Zhao, F., Ai, Z., & Gong, Z. (2014). Parameter tuning of adrc and its application based on cccsa. *Nonlinear Dynamics*, *76*(2), 1185–1194.

Zhang, H., Xiao, G., Yu, X., & Xie, Y. (2020). On convergence performance of discrete-time optimal control based tracking differentiator. *IEEE Transactions on Industrial Electronics*.

Zhang, T., Xu, Z., & Gerada, C. (2019). A nonlinear extended state observer for sensorless ipmsm drives with optimized gains. *IEEE Transactions on Industry Applications*, *56*(2), 1485–1494.

Zhao, Z.-L., & Guo, B.-Z. (2015). On active disturbance rejection control for nonlinear systems using time-varying gain. *European Journal of Control*, *23*, 62–70.