# Differential Evolution Based PID Antenna Position Control System with Disturbance Mitigation

Sunday Iliya[1], Olurotimi Olakunle Awodiji [2], Geraldine Rangmoen Rimven[3]

sundayiliyagoteng@yahoo.com, awodijiio@unijos.edu.ng, daloengg@unijos.edu.ng

**Abstract**— This paper presents a robust and efficient way of tuning PID controller using different variants of differential evolution (DE) algorithms for control of antenna positioning system and external disturbance mitigation. Quite a number of failure in complex control systems e.g. flights is attributed to external disturbance resulting from natural events, and sometimes from man oriented events. Hence controllers should not only be rated based on their ability to track the command input (target), but also in their ability to mitigate the effects of external disturbance. Five DE variants were implemented in this experiment, out of which DE/rand/2/JDE/bin appear to be optimal in addressing the problem, with maximum disturbance amplitude of 0.00073 which decayed rapidly to zero within 3 seconds, peak overshot of 0.0167, rise time of 0.04sec, and settling time of 0.08sec. This has an overall cost or objective fitness function of 0.034. The second optimal optimizer is DE/rand/1/bin with maximum disturbance amplitude of 0.00079.

**Keywords:** —Differential evolution algorithms, PID controller, Step response, Ziegler–Nichols tuning method, optimization, objective fitness function.



## 1. INTRODUCTION

ONe of the major challenge of any positioning control system is the ability to coup with unpredictable disturbance and changes resulting from within the system or its domain. Antenna positioning system are face with different challenges among which is disturbance from natural events such as wind. The aim of this research is to design an intelligent control schemes to position the antenna in the direction of the main lobe for optimum reception/transmission in a dynamically changing environment.

## 2. OPTIMIZATION OR TUNING ALGORITHMS

A brief description of the optimization algorithms implemented are presented in this section. We explore the advantages of global search capability of population based differential evolutionary algorithms variants to evolve the gains of the PID controller. The complexity of many heuristic controllers becomes increasingly complicated due to meta parameters (free parameters) in the model or controller frame work that govern their behaviour and efficiency in optimizing a given problem. How best a given controller can solve a given problem, depends on the correct choice of the meta parameters. The values of those parameters are problem dependent, thus for each problem, those parameters need to be fined tune to get the optimum or near optimum. The tuning pose another optimization problem. The PID gains of the antenna positioning system depicted in this paper were optimized using population based randomization optimization algorithms.

## 3. DIFFERENTIAL EVOLUTION (DE)

DE are population based direct search algorithms used to solve continuous optimization problems. DE aims at evolving NP population of D dimensional vectors which encodes the G generation candidate solutions $X_{i,G}=X^1_{i,G} \ldots X^D_{i,G}$ towards the global optimum, where i=1, … NP. The initial candidate solutions at G=0 are evolves in such a way as to cover the search space as much as possible by uniformly randomizing the candidates within the decision space using Eq (1) [6][7][3].

$$X_{i,G} = X_{min} + rand(1,0).(X_{max} - X_{min}) \quad (1)$$

Where $i = 1, \ldots, NP$, $X_{min} = X^1_{min} \ldots X^D_{min}$, $X_{max} = X^1_{max} \ldots X^D_{max}$ and rand(1,0) is a uniformly distributed random number between 0 and 1.

### 3.1 Mutation

For every individuals (target vectors) $X_{i,G}$ at generation G, a mutant vector $V_{i,G}$ called the provisional or trial offspring is generated via certain mutation schemes [6][7][3]. The mutation strategies implemented in this study are:

DE/rand1

$$V_{i,G} = X_{r1,G} + F.(X_{r3,G} - X_{r2,G}) \quad (2)$$

DE/best/1:

$$V_{i,G} = X_{best,G} + F.(X_{r2,,G} - X_{r1,G}) \quad (3)$$

DE/rand-to-best/1:

$$V_{i,G} = X_{i,G} + F.\left( X_{best,G} - X_{ri,G}\right) + F.(X_{r2,G} - X_{r1,G}) \quad (4)$$

DE/best/2:

$$V_{i,G} = X_{best,G} + F.\left(X_{r2,,G} - X_{r1,G}\right) + F.(X_{r4,G} - X_{r3,G}) \quad (5)$$

Where the indexes r1, r2, r3 and r4 are mutually exclusive positive integers and distinct from i. These indexes are generated at random within the range [1 - PN]. $X_{best,G}$ is the individual with the best fitness at generation G while F is the mutation constant.

### 3.2 Cross Over

After the mutants were generated, the offspring $U_{i,G}$ are produced by performing a crossover operation between the target vector $X_{i,G}$ and its corresponding provisional offspring $V_{i,G}$. The two crossover schemes i.e. exponential and binomial crossover are used in this study for all the DE algorithms implemented. The binomial crossover copied the $j^{th}$ gene of the mutant vector $V_{i,G}$ to the corresponding gene (element) in the offspring $U_{i,G}$ if rand(0,1) ≤ CR or $j=j_{rand}$. Otherwise it is copied from the target vector $X_{i,G}$ (parent). The crossover rate CR is the probability of selecting the offspring genes from the mutant while $j_{rand}$ is a random number in the range [1 - D], this ensure that at least one of the offspring gene is copied from the mutant. If CR is small it will result in exploratory moves parallel to a small number of axes of the decision space .i.e. many of the genes of the offspring will come from its parent than from the mutants, consequently the offspring will resemble its parent. In this way, the DE will serve as a local searcher as it bear strong exploitative capabilities than being explorative. On the other hand, large values of CR will lead to moves at angles to the search space's axes as the genes of the offspring are more likely to come from the provisional offspring (mutant vector) than its parent. This will favour explorative moves. The Binomial crossover is represented by Eq(6).

$$\begin{cases} V_{iG}^{j} \; if \;\; rand(0,1) \leq CR \; or \; j = j_{rand} \\ X_{iG}^{j} \; Otherwise \end{cases} \quad (6)$$

For exponential crossover, the genes of the offspring are inherited from the mutant vector $V_{i,G}$ starting from a randomly selected index j in the range [1 - D] until the first time rand(0,1) > CR after which all the other genes are inherited from the parent $X_{i,G}$[6][7][3].The exponential crossover is as shown in Fig 1.

$U_{iG} = X_{iG}$

$generate = randi(1, D)$

$U_{iG}^{j} = V_{iG}^{j}$

$K = 1$

**while** $rand(0,1) \leq CR$ *and* $K < D$

$U_{iG}^{j} = V_{iG}^{j}$

$j = j + 1$

**if** $j = D$ **then**

$j = 1$

**end** *if*

$K = K + 1$

**end while**

Fig. 1: Exponential crossover



### 3.3 Selection process

When the offspring $U_{iG}$ is birthed via the crossover scheme, to determine whether the offspring should replace its parent XiG or not in the next generation, a greedy selection schemes based on Darwinian Theory is employed. The cost functions (commonly referred to as the fitness functions) $f(U_{iG})$ and $f(X_{iG})$ of the offspring and its parent respectively are computed and compared. If $f(U_{iG}) < f(X_{iG})$ the offspring will replaced its parent in the next generation i.e. $X_{iG+1} = Ui_{G}$ otherwise its parent will be allowed to continue in the next generation $X_{iG+1}=X_{iG}$. This scheme is based on the principles of survival of the fittest. The fitness function used in this research is the weighted sum of the overshot (over or under shot), rise time and the settling time when a unit step input command is used.

### 3.4 jDE

As is always the case in many optimization problems to be solved using heuristic optimization algorithms, the best optimizer for the particular problem is often not known prior to trial. Hence we deem it fit to try one of the famous variant of DE called jDE. The jDE scheme is a popular way of enhancing DE performance with a moderate programming effort. The jDE algorithm enhance the pool of DE search moves by including a certain degree of randomization into the original DE framework. In jDE, the values of mutation and crossover are encoded within each individual candidate solution. For example, the generic individual $X_i$ will be composed of

$$X_i = (X_i[1], X_i[2], \ldots, X_i[D], F_i, CR_i) \quad (7)$$

Hence, at every generation, the offspring is generated for each individual with the parameters $F_i$ and $CR_i$ belonging to its parent. Furthermore, these parameters are periodically refreshed on the basis of the following randomized criterion [6][7]:

$$F_i = \begin{cases} F_L + F_U.rand1 \ \ if\ rand2 < \mathcal{T}_1 \\ F_i \qquad Otherwise \end{cases} \quad (8)$$

$$CR_i = \begin{cases} rand3\ if\ rand4 < \mathcal{T}_2 \\ CR_i \qquad Otherwise \end{cases} \quad (9)$$

Where $rand_j$ are uniform pseudo-random values between 0 and 1. $\mathcal{T}_1$ and $\mathcal{T}_2$ are constant values which represent the probabilities of updating the parameters $F_i$ and $CR_i$ respectively, $F_L$ and $F_U$ are constant values which represent the minimum value that F could take and the maximum variable contribution to F, respectively.

### 3.5 Fitness Function Evaluation

The optimization problem presented in this paper is a multi-objective optimization problem. This is because there are three cost functions to be minimised i.e. the maximum overshot ($M_o$), rise time ($T_r$) and settling time ($T_s$). In order to get a generalized and robust controller gains, the problem is converted to single objective problem with one cost function consisting of the weighted sum of the three objective functions, Eq (10). The weights depends on the important or cost of risk resulting from that particular performance index. This approach is robust because different models can be evolved by just changing the weight to meet up with setting performance specifications or criteria.

$$\gamma = \alpha_o M_o + \alpha_r T_r + \alpha_s T_s \quad (10)$$

Where: $\gamma$ is the fitness function, $M_o$ is the maximum overshot, $T_r$ is the rise time and $T_s$ settling time, while $\alpha_o$, $\alpha_r$, and $\alpha_s$ are their weights respectively. It is a generally acceptable practice that the sum of the weights should be 1 i.e. $\alpha_o + \alpha_r + \alpha_s = 1$. This constrain is heuristics and empirical, hence the setting of the weights is problem dependent and cannot be generalised. For this research, after a repeated manual tuning, the following values were found to be optimal with $M_o$ having the highest priority, $\alpha_o$=0.6, $\alpha_r$=0.2, and $\alpha_s$=0.2. For this application, the maximum value a weight can take is 1.

## 4. PROPORTIONAL PLUS INTEGRAL PLUS DERIVATIVE (PID) CONTROLLER

It is interesting to know that nearly half of the industrial controllers used today are PID or modified PID or derivatives of PID controllers. Some intelligent controllers e.g. Fuzzy logic or adaptive fuzzy logic are derivatives of basic PID i.e. they make use of the error and its derivative (rate of change of the error). There are different variant of the PID controller, the one used in this research is given by equation (11) while the transfer function $G_c(s)$ of the controller is depicted by Eq (12), [2][1][4]. A proportional controller will have the effect of reducing the rise time, but will not eliminate the steady-state error. Because of the present of pole at the origin introduced by the integral controller, the integral control will have the capability of eliminating the steady-state error, but it may make the transient response worse. The derivative control will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. The derivative controller predict future error using the rate at which the error is changing while the integral captured the cumulative effects of past errors to improve the system performance.

$$PID = K_p\left(e(t) + \frac{1}{T_i}\int_{t_0}^{t} e(t)\,dt + T_d\frac{de(t)}{dt}\right) \quad (11)$$

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \quad (12)$$

Where: t is time, e(t) is present error at time t, $K_p$ is the proportional gain while $T_i$ and $T_d$ are integral and derivative time constants respectively, s is Laplace complex notation.

### 4.1 Tuning of the PID gains ($K_p$, $T_i$ and $T_d$) Ziegler–Nichols

The process of selecting the controller parameters $K_p$, $T_i$ and $T_d$ to meet a given performance specifications is known as controller tuning. Different variant of population based differential evolution (DE) algorithms were used to evolve the PID gains. One of the major challenge is to define the decision search space i.e. the range within which each of the meta parameters ($K_p$, $T_i$ and $T_d$) of the controller should be searched. To address this problem, Ziegler–Nicholstuning method was used to obtain the centre of the radius of the search space. The Ziegler–Nichols reference gains were obtained using the mathematical model of the antenna positioning system shown in Fig. (2). The centre of the radius for each of the gains $K_p$, $T_i$ and $T_d$ are given by equations (13), (14) and (15) respectively [2].

$$K_p = 0.6K_{cr} \quad (13)$$

$$T_i = 0.5P_{cr} \quad (14)$$

$$T_d = 0.125P_{cr} \quad (15)$$

Where $K_{cr}$ and $P_{cr}$ are the critical gain and critical frequency for self-sustained oscillation of the system.

The decision search space for each of the gains were obtained as follows:

$$K_{p(space)} = [\alpha_{min}K_p,\ \alpha_{max}K_p] \quad (16)$$

$$T_{i(space)} = [\beta_{min}T_i,\ \beta_{max}T_i] \quad (17)$$

$$T_{d(space)} = [\mu_{min}T_d,\ \mu_{max}T_d] \quad (18)$$

$K_p$, $T_i$ and $T_d$ are given by equations (13), (14) and (15) respectively while after a manual tuning, the minimum and maximum values of α, β and μ were obtained as follows: $\alpha_{min} = 0.4, \beta_{min} = 0.2,\ \mu_{min} = 0.2,\ \alpha_{max} = 5,\quad \beta_{max} = 4, \mu_{max} = 4$

## 5. MATHEMATICAL MODEL OF THE ANTENNA POSITIONING SYSTEM

The rotation of the antenna is achieved using DC motor. The aim is to ensure that the antenna (dish) is in line of side with the main lobe for maximum reception.

$$V = R_a I_a + L_a \frac{dI_a}{dt} + E_b \qquad (19)$$

$$T = J\frac{dw}{dt} + Fw \qquad (20)$$

$$E_b = K_b w \qquad (21)$$

$$T = K_t I_a \qquad (22)$$

$$w = \frac{d\Theta}{dt} \qquad (23)$$

Where V is motor terminal supply voltage, $R_a$ armature resistance, $L_a$ is armature inductance, $I_a$ is armature current, $E_b$ is back emf (electromotive force), T is the torque, w is the angular speed in rad/s, J is the inertia constant while F is the viscose constant, $K_b$ is the back emf constant, t is time and Ɵ is angular position in rad. The block diagram shown in Fig. 2 was obtain using equations (19) to (23) along with the controller, where $\Theta_R$ is the command reference input angle while Ɵ is the actual output.

## 6. RESULTS

Each of the DE variant is run for 600 generations consisting of 10 potential candidate solutions. At the end of the generation, the must fitted (best) candidate is used to set the PID gains. The objective fitness function (cost function) used during the training is the weighted sum of the maximum overshot, rise time and settling time, Eq. (10). The evolved best candidate solution was used to control the antenna positioning system using three different approaches, i.e. the system was tested using standard ram and parabolic input command. Thirdly, a real world scenario was modelled as a command input to demonstrate how the output of the system can track the target input, and at the same time mitigating the effects of external disturbance. The performance index used to evaluate the accuracy of the system in tracking the command in put is the root mean square error (RMSE) given by Eq. (24). It is interesting to note that the fact that the system depicted good performance for standard ram and parabolic input with low RMSE does not necessarily mean that the system will perform optimally when subjected to real world scenario with unpredictable input commands, coupled with disturbance resulting from natural and man oriented sources. This is revealed when the untune controller obtain directly using Ziegler–Nichols method was used. The RMSE of ram and parabolic command using untune PID for a given DE variant shown in Fig. 6 and Fig. 5 are 0.0164 and 0.0598 respectively while for the tuned PID are 0.0212 and 0.0042 respectively. But when the tuned and the untune PIDs were tested using real world command input, the untune PID perform poorly with RMSE of 6.653 while the tuned PID followed the command input closely with RMSE of 1.9611 as shown in Fig. 4. More importantly, the effects of external disturbance which is one of the key cost function in this research, were highly attenuated with maximum disturbance amplitude of 0.00073 which decayed rapidly to zero within 3 seconds as shown in Fig 7. This research also validate that PID gains obtained using Ziegler–Nichols method may not be the optimum but is a valuable tool for obtaining the radius of the search domain within which the optimum or near optimum are likely to be found. The details of the numeric results obtained from the DE variants implemented in this research are shown in table 1. Five DE variants were implemented. Where bin refer to Binomial crossover and exp refer to exponential crossover.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Theta_{Ri} - \Theta_i)^2} \qquad (24)$$

Where: RMSE is the root mean square error, N is the number of simulation time steps, $\Theta_{Ri}$ and $\Theta_i$ are the command input and the actual output at time index I respectively.



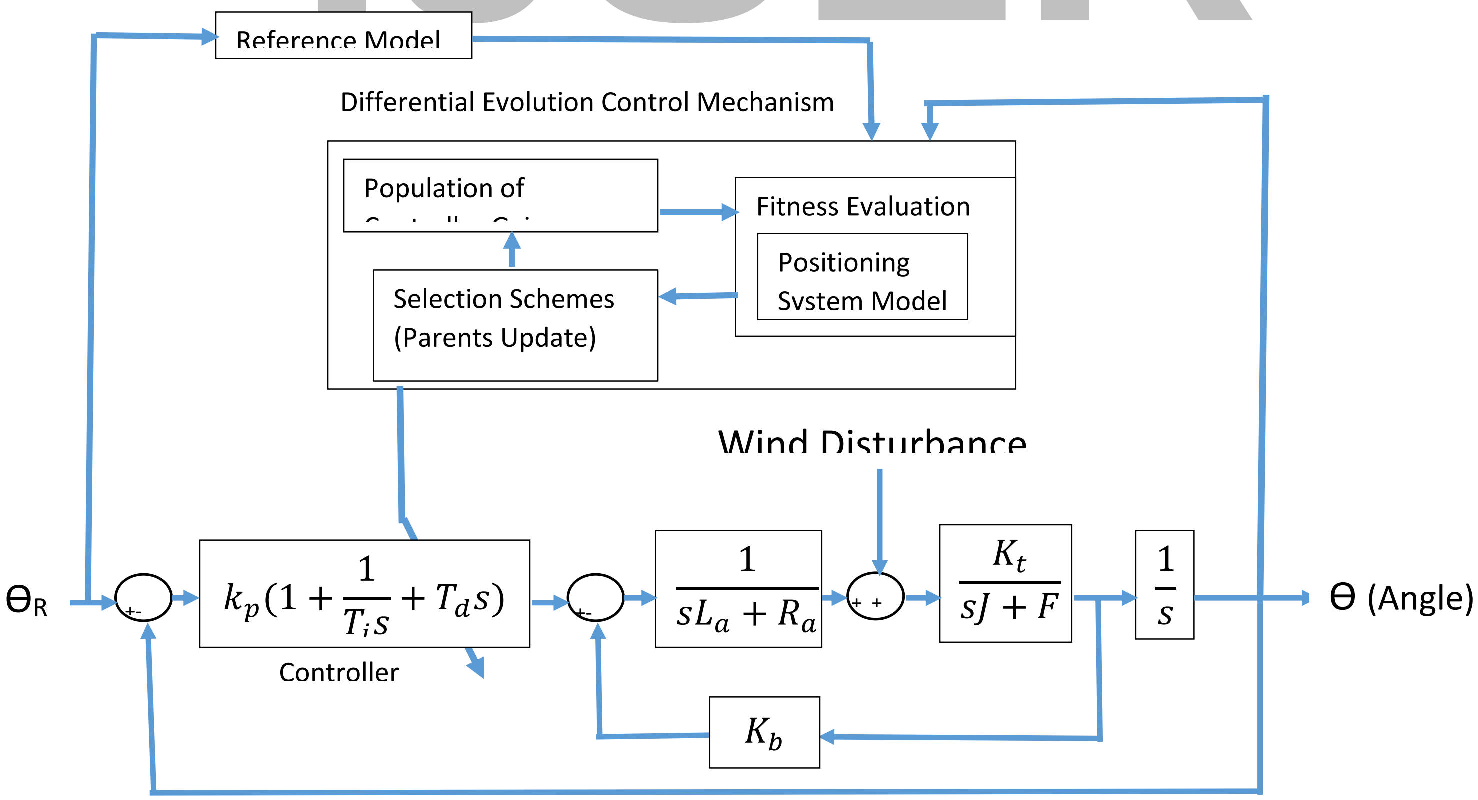


Fig. 2: Block diagram of the antenna positioning system

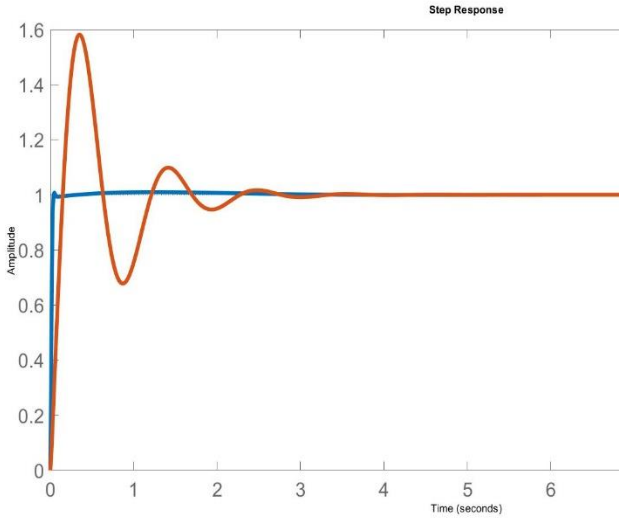


Fig. 3: Unit step response using: tuned PID and untune PID

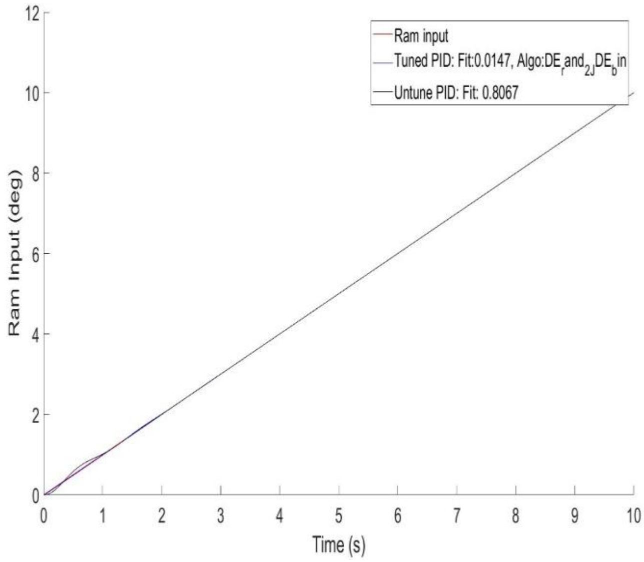


Fig. 6: Ram input command

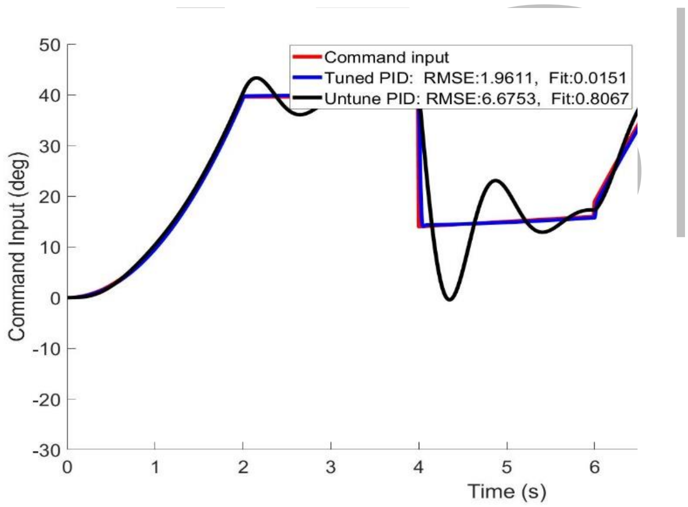


Fig 4: Real world command input using tuned and untune PID

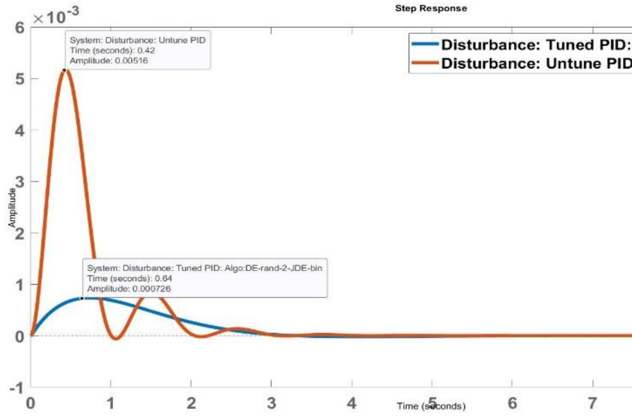


Fig. 7: Disturbance Step Response

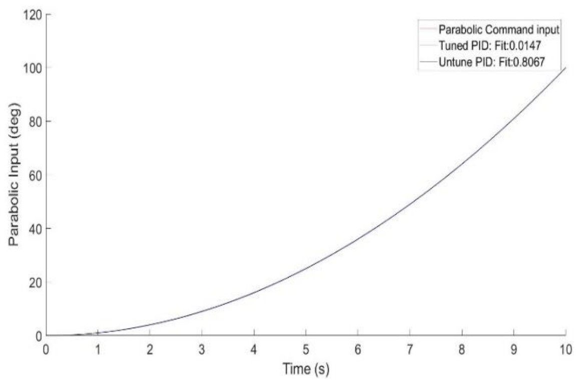


Fig. 5: Parabolic input command

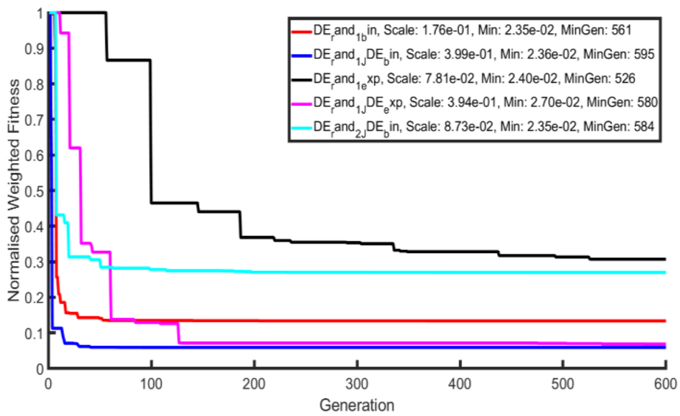


Fig. 8: DE Optimization Generation Fitness Function

Table 1: Performance of the various DE variants implemented

| Algorithms | Command RMSE | Disturbance Max Unit Step Response Amplitude | Max Overshot | Rise Time | Settling Time | Fitness |
|---|---|---|---|---|---|---|
| DE_rand_1_bin | 1.844532561 | **0.00079** | 0.0163 | 0.02 | 0.09 | 0.0318 |
| DE_rand_1_JDE_bin | 2.015262894 | 0.00170 | 0.0128 | 0.03 | 0.04 | 0.0217 |
| DE_rand_1_exp | 3.902446441 | 0.00110 | 0.1832 | 0.11 | 0.46 | 0.2239 |
| DE_rand_1_JDE_exp | 1.838911041 | 0.00120 | 0.0339 | 0.01 | 0.33 | 0.0883 |
| DE_rand_2_JDE_bin | 2.268525662 | **0.00073** | 0.0167 | 0.04 | 0.08 | 0.0340 |

## Conclusion

DE proved to be an efficient optimizer for tuning the PID gains for tracking of the command input (target), and for mitigation of the effect of external disturbance. With **DE/rand/2/JDE/bin** algorithm emerging as the best for mitigation of external disturbance, with maximum disturbance amplitude of 0.00073 followed by **DE/rand/1/bin** with disturbance amplitude of 0.0079. But **DE/rand/1/bin** tract the command input more closely (accurately) than **DE/rand/2/JDE/bin** with RMSE of 1.8445. Hence, the recommended optimizer for this problem that effectively tracked the target command, and also for efficient mitigation of the effect of external disturbance is **DE/rand/1/bin.**

## References


[1] A.R. Laware1, V.S. Banda and D.B. Talange. Real Time Temperature Control System Using PID Controller and Supervisory Control and Data Acquisition System (SCADA),*International Journal of Application or Innovation in Engineering & Management*, *Volume 2, Issue 2, 2013*

[2] I J Narath and M. Gopal: Control system Engineering, fourth Edition, 2006

[3] J. Brest, S. Greiner, B. Boškovic, M. Mernik, and V. Žumer. Self-Adapting ControlParameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation, 10(6):646–657, 2006.

[4] Katsuhiko Ogata Modern Control Engineering *Fifth Edition* Upper Saddle River 3010

[5] K Smriti Rao, Ravi Mishra Comparative study of P, PI and PID controller for speed control of VSI-fed induction moto, *International Journal of Engineering Development and Research*, *Volume 2*, *Issue 2*, 2014

[6] S. Iliya. Application of Computational Intelligence in Cognitive Radio Network for Efficient Spectrum Utilization, and Speech Therapy. PhD Thesis, 2017.

[7] S. Iliya, F. Neri, D. Menzies, P. Cornelius, and L. Picinali. Differential evolution schemes for speech segmentation: A comparative study. In IEEE Symposia Series on Computational Intelligence (SSCI), 2014.