Iran University of Science and Technology

International Journal of Automotive Engineering

Journal Homepage: ijae.iust.ac.ir

ISSN: 2008-9899

Optimal Idle Speed Control of a Natural Aspirated Gasoline Engine Using Bio-inspired Meta- heuristic Algorithms

S.Ali MirMohammadSadeghi [1], Kamyar Nikzadfar [2*], Nima Bakhshinezhad [1] , Alireza Fathi [1]

[1] Babol Noshirvani University of Technology, Mazandaran, Iran

[2] Assistant Prof. Kamyar Nikzadfar, Dept. of Mechanical Engineering, Babol Noshirvani Univ. of Technology, Shariati Ave., Babol, Mazandaran, Iran

ARTICLE INFO

Article history:

Received : 18 July 2018

Accepted: 24 Sep 2018

Published: 30 Sep 2018

Keywords:

PID Controller Tuning

Optimal Control Parameter Optimization Metaheuristics

Mean Value Model (MVM) Engine Control

A B S T R A C T

In order to lowering level of emissions of internal combustion engines (ICEs), they should be optimally controlled. However, ICEs operate under numerous operating conditions, which in turn makes it difficult to design controller for such nonlinear systems. In this article, a generalized unique controller for idle speed control under whole loading conditions is designed. In the current study, instead of tedious time-consuming trial-and-error based methods, soft computing techniques are employed to tune a proportional-integral-derivative (PID) controller which controls idle speed of engine. Since model based design technique is employed, a mean value model (MVM) is taken advantage due to its evidenced merits. Moreover, a brief introduction to the selected meta-heuristics is given followed by a flowchart to show how the engine model is linked to the optimization algorithms. A set point of 750 rpm is fed to the system, and the weighted sum of the three characteristics of mean squared error, control energy, and percent overshoot of the control system is set to the problem objective function to be minimized. It is evidenced that of all the examined meta-heuristics, Bees Algorithm (BA) converges to a better solution. Finally, to consider the effectiveness of the developed optimal controllers in disturbance rejection, they are implemented to the engine MVM model. The results of the research indicate, all the four optimally designed control systems, albeit the intermediate superiority, are of conspicuous success in compensating for the input disturbances of the load torque.

1. Introduction

Internal combustion engines (ICEs) play an important role nowadays in the field of automotive engineering. Notwithstanding the advantages that this ubiquitous part of industry and society can provide, the detrimental issue of air pollution can be considered as a token of their disadvantage. This is where engine control comes in to efficiently optimize the fuel consumption needs and hence minimize the level of emissions released. Without jeopardizing the automotive drivability, control engineers are able to gather both the goals of the efficient fuel consumption and a guaranteed level of emission together, using microelectronic control systems. An electronic throttle valve is composed of a DC motor, a motor pinion gear, an intermediate gear, a sector gear, a valve plate, a nonlinear spring, and a position sensor [1]. A mean value model (MVM) is a mathematical engine model which is intermediate between large cyclic simulation models and simplistic transfer function models. It predicts the mean values of major external engine variables such as crankshaft speed and manifold pressure dynamically in time [2]. Jazayeri et al. (2005) used the MVM technique to achieve the advantages of high calculation speed

*Corresponding Author
Email Address: nikzadfar@nit.ac.ir

[DOI: 10.22068/ijae.8.3.2792]

and consisting a decent accuracy. In order to increase the accuracy of manifold pressure calculations, they proposed and compared two different relations. They have also presented a set of equations to describe rotational dynamics. The accuracy of the developed model has been validated through experimental works conducted on the engine of a Samand automobile [3]. To increase the capability of conventional MVMs in the prediction of engine raw emissions and performance in transient regimes, Nikzadfar and Shamekhi (2015) used an extended mean value model (EMVM) to study control-oriented modeling of ICEs based on block oriented modeling (BOM) concept. Validating the developed model with experimental data, they could achieve a satisfying concurrence between the two [4].

Of all the existing controllers, PID controllers are evidenced to be of a great significance, which makes them be widely used in the industry due to their robust performance, albeit their simple structure. Unfortunately, since many real-world systems are mostly burdened with problems including a high order, time delays, and nonlinearities, it is quite a bit difficult to tune the appropriate gains of PID controllers [5]. Therefore, soft computing-based PID tuning techniques have intrigued researchers from an eclectic realm of science recently [6-9]. Abachizadeh et al. (2010) have employed Artificial Bee Colony (ABC) algorithm to tune PID controller parameters for plants of a high order as well as systems with time delay [9]. Lin et al. (2003) proposed a Genetic Algorithm (GA) based multi-objective PID controller for a linear brushless DC motor [10]. Bagis (2007) has used a modified version of GA to find the optimum PID parameters, and he compared the results with fuzzy and classic methods [11]. Bassi et al. (2011) used Particle Swarm Optimization (PSO) algorithm for optimal tuning of PID controllers, referred to as PSO-Based PID controller. As result of their research indicated, the recruited technique outperforms the traditional rival method of Ziegler-Nichols in improving the system's transient response [12]. Hendricks and Sorenson have developed an engine MVM model to show how the engine model can be applied in order to systematically design and analyze the classical electronic engine control systems [13]. Fiaschetti,and Narasimhamurthi have proposed a descriptive bibliography about modeling and control of SI, 4-stroke engines; moreover, the discussion of the fundamentals in this work paves the way for uninitiated researchers in related areas [19]. Weeks and Moskwa have presented a Matlab/Simulink model which simulates a port-fuel-injected, spark-ignition engine. In addition, it includes the dynamics of fuel, air, and EGR in the intake manifold in a four-stroke engine [27].

In accordance with the previous investigations, ICEs are evidenced to be high nonlinear systems. That is to say, a controller that is designed for one operating condition may not function properly for another, i.e. the stability of the closed-loop system may even be jeopardized. In order to extricate from this flaw, a generalized controller must be designed such that it performs appropriately at all the engine operating points. In other words, due to the high variety of conditions at which an engine operates, designing a PID controller with constant gains that can generally function at all the operating points may become vital. Aiming at doing so, in the current investigation, in order to control the engine idle speed, an ICE is modelled taking advantage of MVM concepts. In the controlled system, the manipulated parameter is the angle of throttle plate (θ). Since the engine model contains many nonlinearities, it is essential to use powerful meta-heuristic algorithms to tune PID controller parameters.

What follows in this article is organized as: In the next section, the modeling equations are governed; afterward, a brief introduction to PID controllers is given. Thereafter, meta-heuristic algorithms are introduced. Moreover, an overview for each of the existing algorithms is given. Besides, the optimization algorithms are implemented to tune PID controller parameters. Finally, after the simulations and computations, the results are compared and concluded.

[DOI: 10.22068/ijae.8.3.2792]

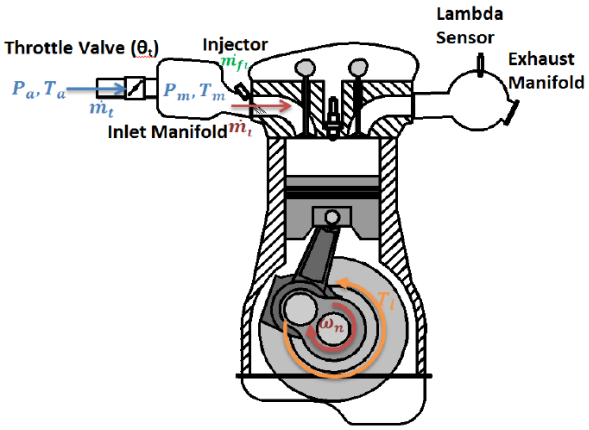

Figure 1 :Schematic of an SI engine cylinder

2. Mean Value Modeling and the Governing Equations

An SI engine is a thermal machine, which converts the contained chemical energy in a mixture of air and fuel into mechanical energy. Conversion of chemical energy to mechanical energy involves some dynamic processes. Take the example of the reciprocating nature of piston synchronized with alternating valve dynamics, which causes a potential for flow of air from ambient into the cylinder. A schematic of the engine cylinder is illustrated in Figure 1; where $\mathrm{P_a} = 100000$ [Pa] and $\mathrm{T_a} = 300$ [K] are pressure and temperature of ambient air, respectively. The parameters of the following equations are obtained from the cited references such that they provide an optimal quality, e.g. minimum level of emissions, minimum fuel consumption, etc. Therefore, some of the parameters are not deeply discussed, for the sake of conciseness, and are instead cited to references; hence, the readers are encouraged to check the corresponding references for further elaboration.

The air is induced through throttle into inlet manifold after which goes through inlet valves to the cylinder. Since the inlet and exhaust air flow-rate are different from each other, there is a dynamic process in the manifold, called manifold process, which is described by [22-24]

$$\dot{P_m} = \frac{RT_m}{V_m}(\dot{m}_t - \dot{m}_\iota) \tag{1}$$

Where P_m is intake manifold pressure, T_m is the inlet manifold gas temperature, which, here, it might be assumed to be equal to temperature of the ambient air. $R = 287\ {}^{\mathrm{J}}/_{\mathrm{kg.\,K}}$ is air constant and $\mathrm{V_m} = 1.55$ Lit is inlet manifold volume. Also $\dot{m}_t$ and $\dot{m}_\iota$ are airflow through the throttle valve and inducted air to cylinder respectively. The air flow rate through throttle is controlled mainly by throttle plate angle (θ) and is calculated as follows [24].

$$\dot{m}_t = \frac{P_a}{\sqrt{RT_a}}\beta_1(\theta)\beta_2(P_r) \tag{2}$$

Where β_1 and β_2 represent, respectively, the effects of the throttle plate angle and the pressure ratio ($P_r = \frac{P_m}{P_a}$) [24]

$$\beta_1(\theta) = 62.56 + 0.67\cos(\theta) - 62.8\cos^2(\theta)$$

$$\beta_2(P_r) = \begin{cases} \dfrac{1}{0.74}\sqrt{{P_r}^{0.4404} - {P_r}^{2.3086}} & P_a \geq 0.4125 \\ 1 & P_a < 0.4125 \end{cases} \tag{3}$$

Also, the induced air in the engine is calculated as follows:

$$\dot{m}_\iota = \frac{V_d}{2RT_m}(0.921P_m - 8200)n \tag{4}$$

Where $V_d = 1.8$ Lit is displacement volume; n is engine rotational speed in [rpm].

The injectors are responsible for injection of fuel into the air stream to form an appropriate air and fuel mixture. Not all the quantity of injected fuel is mixed with air due to evaporation process. A main dynamic engine relates to a process of injection and evaporation. The manipulated parameter for injection is injection flow rate ($\dot{m}_\psi$) while the quantity of fuel which is conducted to cylinder is $\dot{m}_f$. Always part of injected fuel adheres on manifold walls called wall-film fuel, which has a mass of ($\dot{m_f}(t)$). The following dynamic equations might be used to model the fuel dynamics [3, 14, 22].

[DOI: 10.22068/ijae.8.3.2792]

$$\dot{m}_f = (1-k)\dot{m}_\psi + \frac{m_{f(t)}}{\tau} \quad (5)$$

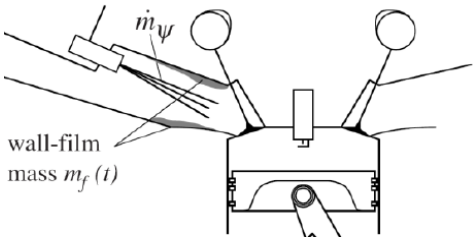

Figure 2 : Fuel injection dynamic process

$$\frac{dm_f}{dt} = k\dot{m}_\psi - \frac{m_{f(t)}}{\tau} \quad (6)$$

Where τ is a time constant and variable k describes how much of the injected fuel adheres on the wall which are, respectively, equal to 0.34 (s) and 0.3. The inducted air fuel mixture are ignited using a spark plug, causing the involved energy to increase the enthalpy of trapped air which in turn increase the pressure of air and move piston downward ending at generation of mechanical power on crankshaft end. The indicated torque of and engine might be calculated using following equation [26-28].

$$T_i = 531500 * \frac{\dot{m}_\iota}{n}\eta_f * SI * AFI \quad (7)$$

in which η_f is the fuel conversion efficiency, SI is spark advanced effect and is equal to unity (SI=1), and AFI is a function of λ as follows:

$$AFI = 48.08\lambda^5 - 235\lambda^4 + 456\lambda^3 - 441.5\lambda^2 + 214\lambda - 40.59 \quad (8)$$

where λ is calculated as follows:

$$\lambda = \frac{\dot{m}_\iota/\dot{m}_f}{14.7} \quad (9)$$

Not all the generated torque in thermodynamic cycle (indicted torque, T_i) might be delivered to crankshaft due to some frictions. The friction torque, which is mainly caused by mechanical friction of moving parts and some pumping losses, can be obtained as follows [29].

$$\frac{T_f}{0.14} = 50.26 + 0.0197N + 1.087 * 10^{-6}N^2 + \frac{17689}{N} + 0.0193\sqrt{N} + 8.23 * \frac{P_m}{P_a} 2.632 * \frac{P_m}{P_a} 9.3^{(1.33-0.00152N)} + 1.3 * 10^{-6}\left(\frac{P_m * N}{P_a}\right)^2 \quad (10)$$

where N is engine rotational speed in "RPM". Finally, the engine rotational speed is calculated as the third dynamic parameter based on Euler equation as follows:

$$\dot{\omega} = \frac{1}{J_e}(T_i - T_f - T_l) \quad (11)$$

where J_e is engine rotational inertia and ω is engine rotational speed in [rad/sec]. T_l is the load torque, which is exerted to engine due to transmission system [3]. In this study, variations of T_l play role as input disturbance over engine rotational speed. The governed equations are modeled using Matlab/Simulink. Figure 3 illustrates the block diagram of the engine system. Why, one has to ask, is the throttle plate controller merely considered to be optimally tuned? Based upon the cited references, from which the governing equations are taken, the injection quality, the injection timing, and ignition timing, which highly dominate the degree to which emissions are released, are set fixed at an optimum point such that they guarantee a minimum level of exhaust emissions.

3. Overview Of PID Controllers

A PID controller is used for improving the dynamic response as well as eliminating/reducing the steady-state error of a controlled system. Adding a finite zero to the open-loop plant transfer function, the derivative controller improves the transient response. The integral controller, however, adds a pole at the origin, which leads to an increase in system type by one and a reduction in the [5]:

$$C(s) = K_p + \frac{K_i}{s} + K_d\frac{N}{1+N\frac{1}{s}} \quad (12)$$

where K_p, K_i, and K_d are, respectively, the

[DOI: 10.22068/ijae.8.3.2792]

proportional, integral, and derivative gains. Moreover, $N = 100$ is the filter coefficient.

discouraged, unless you have verified that they will be understandable when printed in black ink.

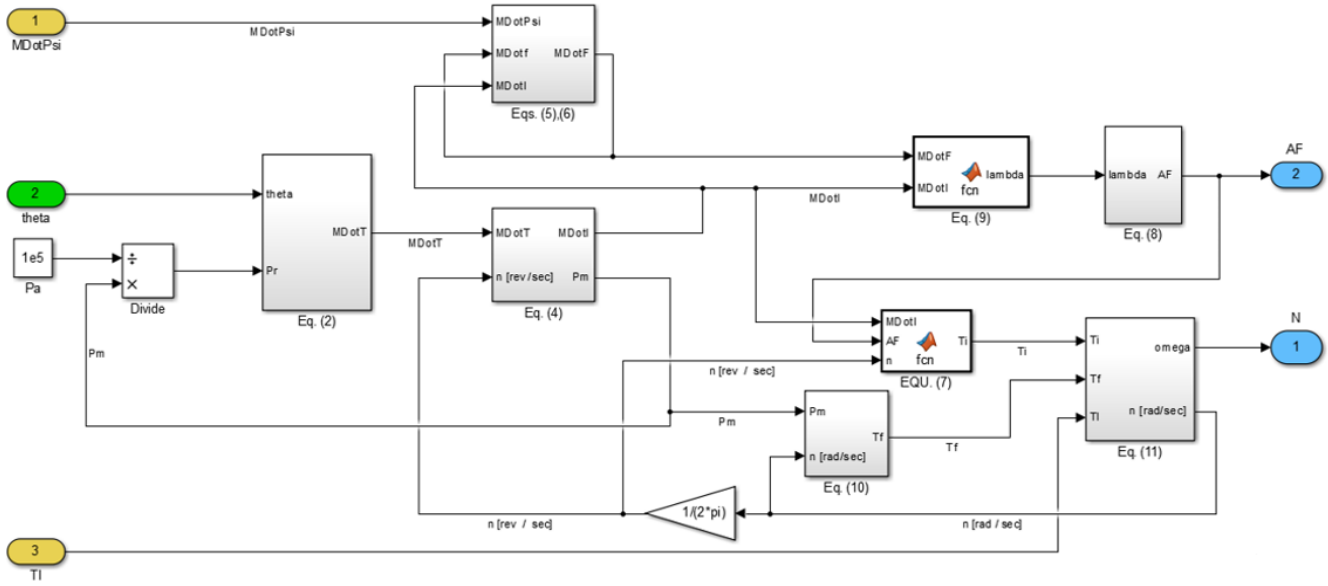

Figure 3: Block diagram of the engine system

4. Introduction to Meta-heuristic Algorithms

Meta-heuristic algorithms are population-based techniques that begin with a set of random solutions and gradually try to converge to a better feasible solution due to the associated mechanism defined for each of the algorithms. These algorithms are typically composed of five major parts. The first part is called *problem definition* where the objective function, number of decision variables, the range of the decision variables, etc. are defined. The second part is *algorithm setting parameters* which specifies the controlling parameters of the selected algorithm; such as the number of population, the maximum number of iterations, and so on. The third part, the *initialization*, is dedicated to the problem initializing. That is to say, a set of random solutions are selected and then the fitness values of each is evaluated. The fourth part is *algorithm main loop*, which, as mentioned before, varies due to the type of the algorithm selected and the mechanism associated. The results are analyzed in the last part, i.e. *post processing* part. Meta-heuristic algorithms can be mainly divided into two categories: evolutionary algorithms (EAs) and swarm intelligence base algorithms.

Place figures, tables, and photographs in the paper near where they are first discussed, rather than at the end, if possible. Color illustrations are

Larger figures and tables that will need the whole width of the page should be located at the top or at the bottom of the page in order to stop interfering with the normal flow of the text as in the example Figure 2.

5. Overview of the Discussed Algorithms

5.1 Genetic Algorithm

Genetic algorithms (GAs) are among stochastic evolutionary-based algorithms inspired by Charles Darwin Theory of Survival of the Fittest and is proposed by Holland (1975) [30]. Care is needed not to restrict the application of GAs only to function optimizers, i.e. they are originally developed to study a much broader realm of problems consisting many non-optimization applications [14,15]. However, GAs are powerful optimization algorithms that reach an approximate global maximum/minimum of complex multivariable functions over a wide search space [33].

The mechanism procedure can be explained as follows:

At the first stage, initialization stage, GAs begin by selecting a number of random solutions (chromosomes) and the fitness values of each are then evaluated. At the second stage a selection strategy – tournament, roulette wheel, or random selection – is used and two number of parents are selected which are going to be crossed over and mutated. Thereafter, the produced offspring

[DOI: 10.22068/ijae.8.3.2792]

population is added to the parents' population. At the third stage, the merged populations pass through sorting and, then, truncation stages. Owing to these last two stages that are applied to the total merged populations, there is no longer a need to be concerned whether the offspring offer a better solution than their parents or not. The result is a population of the same size as before offering a better solution compared to the last population. Finally, at the fourth stage, one can select the first individual as the best answer within each iteration. Stages 2, 3, and 4 are iteratively repeated until a termination criterion is met. In this paper, for the selection stage, the Roulette wheel selection process is implemented and the selection probability can be obtained using Boltzmann distribution equation:

Table 1 GA setting parameters

Symbol	DEFINITION	Value
Max_{iter}	Maximum number of iterations	50
N_p	Number of population	10
C_P	Cross over percentage	0.8
M_P	Mutation percentage	0.3
μ	Rate of mutation	0.02
βv	Selection pressure	8

$$p_i = \frac{e^{-\beta/F_i}}{\sum_{j=1}^{N_p} e^{-\beta/F_j}} \quad (13)$$

where p_i is the selecting probability of the i^{th} for the mating pool; F_i is fitness the i^{th} chromosome in the population of size N_p [34].

In this study, GA is mainly used to find to three optimal controller parameters: K_p, K_i, and K_d such that the controlled system obtain an acceptable step response output as well as a good steady-state error. The GA setting parameters are chosen as is shown in Table 1.

5.2 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is among swarm-based EAs which has been extensively applied to solve many sophisticated engineering problems. The PSO algorithm is a bio-inspired algorithm which mimics the natural behavior of schools of fishes, flocks of animals such as birds, insects, and so on [11]. This algorithm is proposed by [35]. The PSO algorithm stages can be explained as follow [36]:

- Assign a random position in the search space to each particle, Initialization stage.
- Evaluate the fitness of each of the particles.
- For each of the particles, compare the particle's current position fitness value with its p_{best}. If the current value is better than p_{best}, substitute p_{best} with the current position; afterward, for each of the particles, compare p_{best} with g_{best}. If the current p_{best} is better than g_{best}, substitute g_{best} with p_{best}.
- Update each particle's position and velocity with respect to their p_{best} and the g_{best}.
- Repeat stages 3 and 4 until a termination criterion is met.

where p_{best} refers to personal best position (candidate solution), and g_{best} refers to global best position (the problem solution ever found).

In this method, the position and velocity of each particle at t^{th} iteration will be updated by the Eqs. (14) and (15), respectively:

$$x_{t+1}{}^i = x_t{}^i + v_{t+1}{}^i \quad (14)$$

$$v_{t+1}{}^i = w_i \times v_t{}^i + c_1 \times rand1 \times \left(p_t{}^i - x_t{}^i\right) + c_2 \times rand2 \times \left(p_t{}^g - x_t{}^i\right) \quad (15)$$

where the subscript g refers to global best; $rand1$ and $rand2$ are two random numbers within [0, 1]; w is the inertia weight [37]; c_1 and c_2 are two constants called, respectively, personal and global learning coefficients. Through a rigorous research, Clerc and Kennedy (2002) have obtained the best numerical values for w, c_1, and c_2 with the aid of constriction coefficients [38]. These factors are respectively proposed to be $w = 0.7298$, $c_1 = 1.4962$, $c_2 = 1.4962$. The parameters of PSO algorithm are set as shown in Table 2.

[DOI: 10.22068/ijae.8.3.2792]

Table 2 PSO setting parameters

Symbol	Definition	Value
Max_{iter}	Maximum number of iterations	50
N_p	Number of population	10
w	Inertia weight	0.7298
c_1	Personal learning coefficient	1.4962
c_2	Global learning coefficient	1.4962

5.3 Invasive Weed Optimization

Invasive weed optimization (IWO) algorithm is a technique inspired by colonizing weeds proposed by A.R. Mehrabian and C. Lucas. As the IWO algorithm is tried to exactly mimic weeds' natural properties of optimality, robustness, and adaptation, it can be concluded that the IWO algorithm is a powerful though simple algorithm. The feasibility, efficiency, and effectiveness of IWO algorithm are well proved in details and the results are also compared to that of some other evolutionary-based algorithms. This algorithm is mainly composed of four major stages: population initialization, reproduction, spatial dispersal, and competitive exclusion. These four major stages are stated as follow [39]:

- A finite number of seeds are dispreaded over the search area (population initialization).
- Each one of the seeds grows into a flowering1 plant and depending on the fitness that they can provide, they themselves produce a number of seeds (reproduction).
- The produced seeds are then randomly dispreaded near to the parent plant over the search area (spatial dispersal).
- Stages 2 and 3 are iteratively continued until a maximum number of plants is reached. Thereafter, as the total number of plants is limited, ineluctably, those plants offering the best fitness can only survive and produce seeds (competitive exclusion). Finally, the process continues until a termination criterion is met.

In reproduction stage, the number of seeds which each of the plants is allowed to produce depends on their own fitness compared to the colony's lowest and highest fitness: the number of seeds that each plant produces is directly proportional to its fitness. In other words, as the fitness of a plant increases, the number of seeds that this plant can produce increases linearly from minimum possible seed production to its maximum. Figure 4 illustrates the above-mentioned procedure.

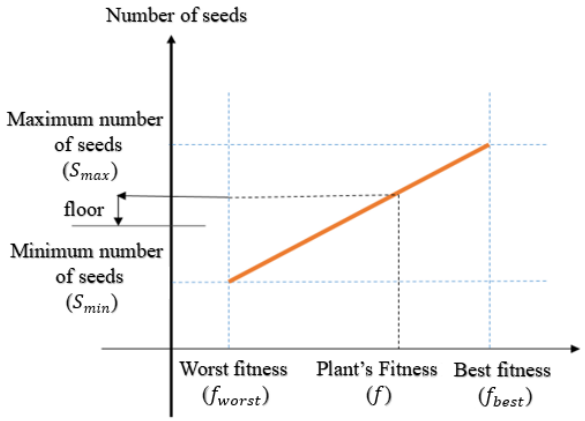

Figure 4*:* Seed production procedure

The number of seeds produced regarding the plant's fitness can be obtained using linear interpolation formula:

$$S = \left\lfloor S_{min} + (S_{max} - S_{min}) * \frac{f - f_{worst}}{f_{best} - f_{worst}} \right\rfloor \quad (16)$$

where S_{min}, S_{max} are respectively minimum and maximum number of seeds; f_{worst}, f_{best}, are respectively the worst and the best fitness; finally, S and f are respectively the number of seeds and the plant fitness. In spatial dispersal stage, the produced seeds are randomly distributed over the i dimensional search space by normally distributed random numbers with mean equal to zero and a variance of σ^2 (Eq. (18)). However, standard deviation (SD), σ, of the random function is a reducing function from an initial value, $\sigma_{initial}$, to a final value, σ_{final}.

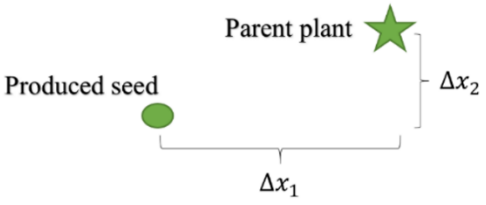

Figure 5 :Seed distribution in a 2-dimensional search space

[DOI: 10.22068/ijae.8.3.2792]

$$x_i^{seed} = x_i^{parent} + \Delta x_i \quad (17)$$

$$\Delta x_i \sim N(0,\ {\sigma_{iter}}^2) \quad (18)$$

$$\sigma_{iter} = \left(\frac{iter_{max} - iter}{iter_{max}}\right)^n * \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \quad (19)$$

where $iter_{max}$ is maximum number of iterations, $iter$ is the current iteration, and n is nonlinear modulation index which controls the rate at which σ_{iter} decreases. The IWO algorithm parameters are set as shown in Table 3.

Table 3 IWO setting parameters

Symbol	DEFINITION	Value
Max_{iter}	Maximum number of iteration	50
N_{init}	Initial number of weeds	5
N_{max}	Maximum number of weeds	10
n	Nonlinear modulation index	2
$\sigma_{initial}$	Initial Value of Standard Deviation	1
σ_{final}	Final Value of Standard Deviation	0.001
S_{min}	Minimum number of seeds	0
S_{max}	Maximum number of seeds	5

5.4 Bees Algorithm

Bees algorithm is a rather simple though highly efficient optimization algorithm. In this method, every point of the search space is thought of as a food source. When a bee visits a food source, it evaluates its quality (fitness). Scout bees measure the fitness space randomly, exploring new areas of high fitness. That is to say, new solutions are randomly explored and evaluated. The visited sites are ranked, and forager bees are hired to exploit the neighborhood of the highest ranking sites. As a result, new solutions near to the best sites are evaluated. The neighborhood of a candidate solution is called a "flower patch". The Bees Algorithm explores the solution space, as an explorative action, and measures the neighborhood of the highest fitness candidate solution, as an exploitative action, looking for the global exterma [32]. The controlling parameters of BA are tabulated in Table 4. demonstrates a flowchart of how BA links to the PID controller tuning.

Table 4 Bees algorithm setting parameters

Symbol	Definition	Value
Max_{iter}	Maximum number of iteration	50
N_{scout}	Number of scout bees	10
N_{sites}	Number of selected sites	5
N_{elite}	Number of selected elite sites	4
N_{bee}	No. of Recruited Bees for non-elite Sites	5
N_{el_bee}	No. of Recruited Bees for Elite Sites	10
r	Neighborhood radius	0.2
r_{damp}	Neighborhood radius damp rate	0.99

6. Results and Discussion

As can be seen from Figure 6, the system consists of a PI controller and a PID controller. On the one hand, the PI controller controls the air-fuel ratio, desired input, which is set to a constant unit step. The injection flow rate, $\dot{m}_\psi$, is then fed forward to the system as the input control signal. The PI controller parameters, i.e. K_p and K_i, are initially adjusted to a pre-defined value of unity. On the other hand, the PID controller controls the engine idle rotational speed, i.e. the desired input, which is set to a constant value of 750 rpm. Needless to say, during the process of tuning the PID controller parameters, $K_p, K_i,$ and K_d, are not constant and vary as functions of the objective values. Obviously, the dynamic behavior from throttle plate angle to the engine rotational speed is more severe compared to that from the applied electric current to the plate angle of the electronic throttle. Therefore, simulation of the rapid dynamic of the electronic throttle is refused.

The connection of the engine Simulink model to the optimization algorithms is briefly analyzed next. According to the Figure 6, the simulation procedure is as follows. First, an optimization algorithm, BA as is indicated, is run. Afterward the

[DOI: 10.22068/ijae.8.3.2792]

algorithm parameters are set, initialization section starts playing its role. In it, the model is run for each random PID gain sets. The mean squared error (MSE) resulted from the error between the set point and actual engine's rotational speed in conjunction with the control energy of the control effort signal are calculated. Added together, these quantities will give the fitness value of the associated candidate solution. While adding to the MSE, we must multiply the control energy by a factor in order to balance the weight of the two objectives. The output of the model, the objective function in here, is the performance index; and its value, now termed as fitness value, is fed back to the algorithm. While initialization, the previously mentioned process is reiterated in number of population size, i.e. N_{p}, and the corresponding fitness values are saved. Next, the algorithm main loop begins. Depending on the algorithm used and the associated population generation mechanism, each newly generated candidate solution is again inserted to the controller and the corresponding fitness value, as is shown in Fig. 7, is then sent back to the algorithm. Thereafter, the particle with the best fitness value is selected as the best particle within each iteration loop. The previously mentioned cycle is repeated until a termination criteria be met. Mean squared error (MSE) between two signals yields the area between the two. Therefore, the lower the MSE between two signals, the closer the two signals will be. Additionally, a lower value for over/undershoot offers a more rapid response and, hence, an improved transient response. The minimization of these two will obviously provide a decent response; yet, there is one other thing needs to be paid attention. The mere minimization of the two will result in a severe demand for the control energy. Thus, this is where the control energy plays role as the other objective to be minimized. In sum, for the set point of 750 rpm to the engine rotational speed, the weighted sum of MSE between the set point and the system output, the percent overshoot (%OS), and the energy of the control effort is considered as the problem objective function, which is given as follows:

$$J = \frac{1}{n}\sum_{i=1}^{n} {e_i}^2 + w * \frac{1}{2}\sum_{i=1}^{n} {u_i}^2 * \Delta t_i + \%OS \quad (20)$$

where n is the error matrix size and e_i is the error signal; $w = 10^{-5}$ is a weight-leveling factor which makes the optimization process be accomplished more indiscriminately towards the other objectives. For all of the algorithms, the lower and upper bound of the search space are set as [0, 100], [0, 100], and [0, 10] for the three decision variables of $K_p, K_i,$ and K_d, respectively. In the current investigation, the algorithms are run with small population size in order to avoid computation obfuscation and to let the solution be converged at a faster rate. In addition, the simulations were conducted in a laptop computer with "Intel Core i5 CPU" at 2.70 GHz and 6GB RAM. Equally important is the fact that all the optimization algorithms have been written in MATLAB script files, i.e. m-files. The simulation results are summarized and tabulated in Table 5. Accordingly, the MSE, Control Energy, and the Percent Overshoot associated with each of the controllers are tabulated. In the meanwhile, in order to select the best PID controller among all the tuned ones, the objective values have been normalized to the range [0, 1] with respect to the maximum value of each column, i.e. the last three columns in Table 5. Consequently, the sum of the normalized characteristics (objectives) was considered the fitness of the corresponding PID controller. As can be seen from the table, of all the tested controllers, BA-PID has the least MSE as well as the least Percent Overshoot, but it offers the largest Control Energy. In addition, as is shown in Figure 7, this algorithm outperforms its rivals and converges to an overall better solution, compared to the other tested algorithms.

[DOI: 10.22068/ijae.8.3.2792]

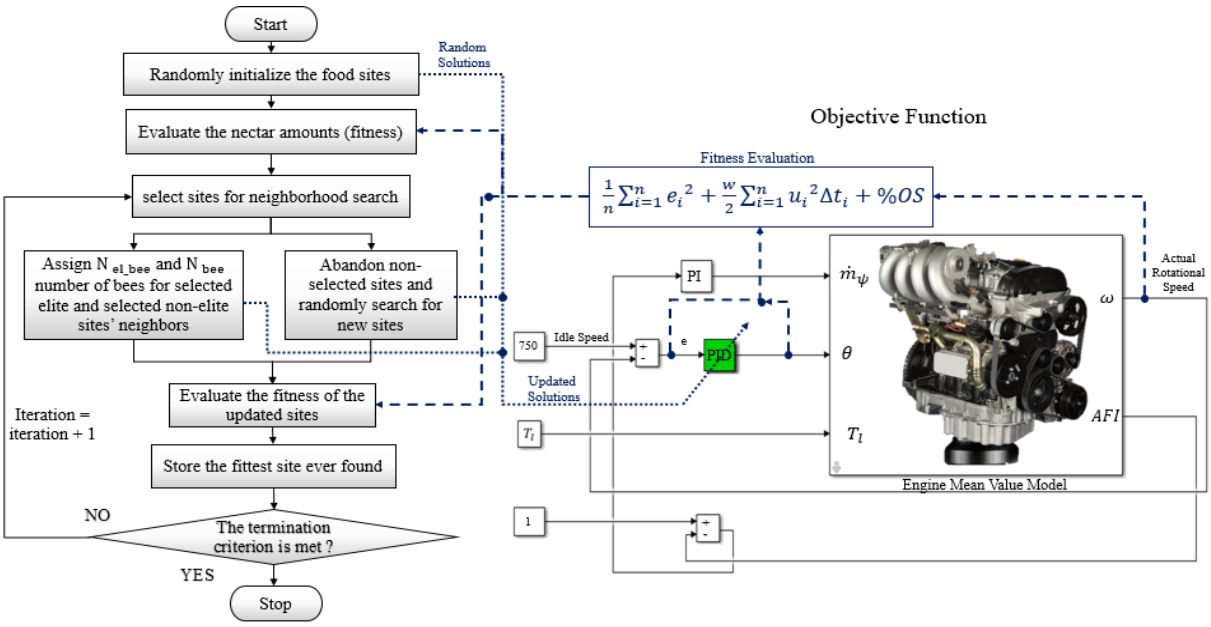

Figure 6 : BA algorithm flowchart for optimal PID tuning

Table 5 The obtained results

Method	K_p	K_i	K_d	MSE	Control Energy	%OS
BA-PID	*5.913897653197346*	*80.9012556397419*	*.202609821644536*	0.0453	338.9526	0.0054
GA-PID	*5.2697843444065*	*99.9306931841722*	*.180691349226107*	0.0468	338.6169	0.0106
PSO-PID	*4.87904705435673*	*74.418306184609*	*.180198642964550*	0.0606	338.6068	0.0099
IWO-PID	*4.3910541044604*	*98.6195822674810*	*0.16978349457987*	0.0571	338.5817	0.0182

Besides, the sum of the normalized objective values were sorted in an ascending order. Figure 8 depicts a bar chart of the normalized values of the objectives related to the four designed controllers. According to Figure 8, the normalized values of the objectives are written on the corresponding bars. Moreover, the controllers' order is organized as sum of the normalized objective values ascends. That is to say, in terms of fitness, the optimally tuned controllers can be sorted from best to worst as follows: BA-PID, GA-PID, PSO-PID, and IWO-PID.

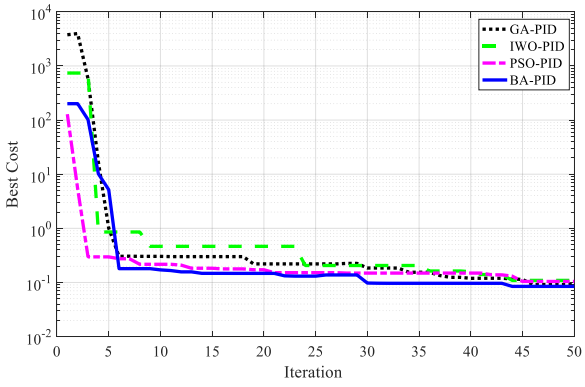

Figure 7: Convergence comparison of GA, PSO, IWO, and BA in minimizing the objective function

[DOI: 10.22068/ijae.8.3.2792]

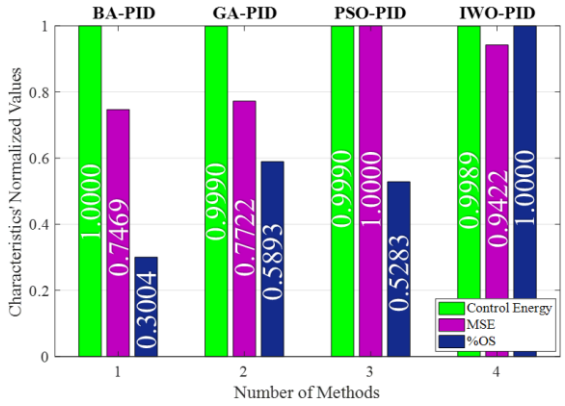

Figure 8: Bar chart of the designed controllers: sorted as sum of the characteristics' normalized values ascends

Now that the controllers are designed, they will be applied to the engine MVM model so that their effectiveness in disturbance rejection be validated. Illustrated in Figure 9, is the implementation of the optimally tuned controllers while compensating for input disturbance of load torque, which increases by time. Also indicated is the control effort signal, i.e. throttle plate angle (θ).

It can be inferred from the figure that all the four optimal controllers, albeit the superiority within, are of conspicuous significance in compensating for the load torque disturbance.

It might be important to note that these slight load disturbances are usually caused by a headwind blowing against the vehicle's travel direction, switching on the vehicle air conditioning system abruptly, and other reasons causing an immediate functioning of the vehicle's dynamo.

7. CONCLUSION

Internal combustion engines (ICEs) are among the most broadly used components of the automotive industry. However, a controller is necessary to maintain the engine at an optimum operating condition so that it yields a minimum level of emissions as well as efficient fuel consumption. In this paper, an internal combustion engine (ICE) was modelled taking advantage of Mean Value Modelling (MVM) concepts.

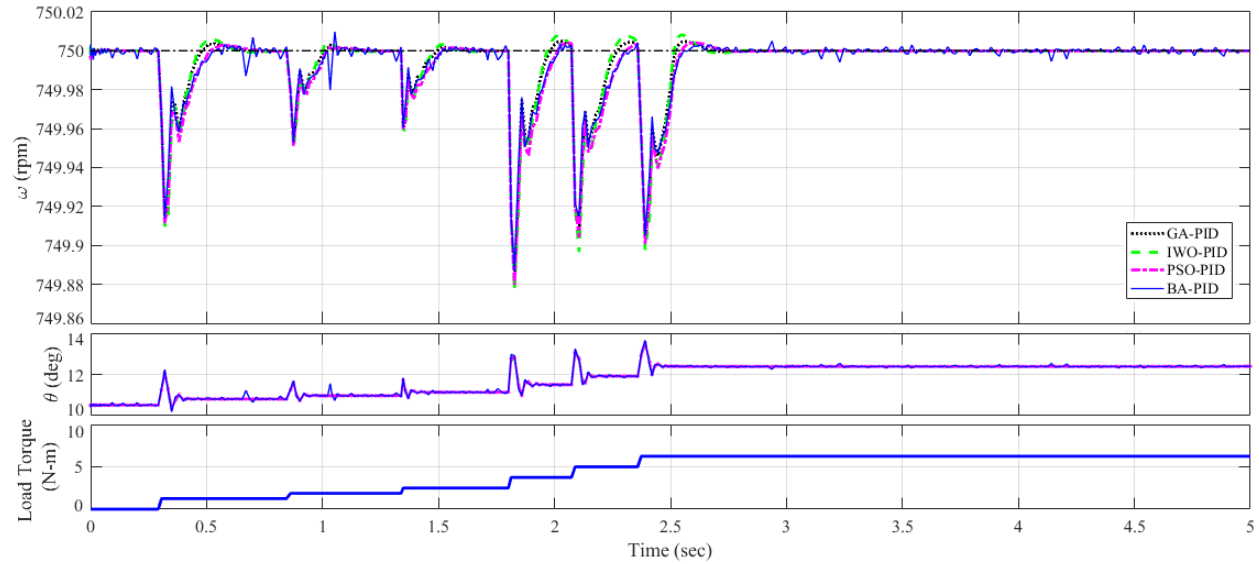

Figure 9: Comparison of step response and transient response improvement of the controllers and their ability to reject input disturbances

[DOI: 10.22068/ijae.8.3.2792]

Taking the throttle valve plate angle (θ) as the controlling parameter, the authors set a PID controller to control the engine idle speed at the reference value of 750 rpm. Thereafter, soft computing-based techniques were recruited to optimally tune the PID controller. Having set a reference signal of 750 rpm to the engine MVM model, the authors formed the cost function of the problem as the weighted sum of the MSE of the error signal, the percent overshoot of the closed loop system response, and the control energy of the control effort. By running the control system via the four tuned PID controllers, the authors have calculated the mentioned objectives associated with the response of each metaheuristic-control system. The sum of the normalized objectives was, then, considered as the fitness of the corresponding PID controller. As a result, the BA-PID controller was shown to be in superiority among the other three rival algorithms by converging to a better solution. At last, the achieved optimal controllers were implemented on the engine MVM model so that their effectiveness be validated in disturbance rejection. As it was evidenced, the recruited optimal metaheuristic-PID control technique exhibited a remarkable adequacy in compensation of the input disturbance of load torque. There are a large number of interesting works to be conducted for further expansion of the current study in the future. For instance, from the optimization perspective, using other metaheuristic algorithms, e.g. Ant Colony Optimization (ACO); Covariance Matrix Adaptation Evolutionary Strategy (CMAES); Teaching-Learning Based Optimization (TLBO); etc., and performing a statistical analysis, e.g. Markov Chain Monte Carlo (MCMC), in order to judge more fairly about the algorithms performance. On the other hand, from the modeling and simulation perspective, the rapid response of the other components of the engine, e.g. the rapid dynamics of the "inside elements" of an electronic throttle, may be taken into account to approach a more real-world simulation.

List of symbols.

ABC	**Artificial Bee Colony**
BA	Bees Algorithm
BOM	Block Oriented Modeling
EA	Evolutionary Algorithm
EMVM	Extended Mean Value Modeling
GA	Genetic Algorithm
ICE	Internal Combustion Engine
IKCO	Iran Khodro Company
IWO	Invasive weed Optimization
MSE	Mean Squared Error
MVM	Mean Value Modeling
OS	Over Shoot
PI	Proportional-integral
PID	Proportional-integral-derivative
PSO	Particle Swarm Optimization
RPM	Revolution per Minute
SD	Standard Deviation
SI	Spark Ignition

References

[1] Pan, Y., Ozguner, U. and Dagci, O.H., 2008. Variable-structure control of electronic throttle valve. *IEEE Transactions on industrial electronics*, *55*(11), pp.3899-3907.

[2] Rajamani, R., 2011. *Vehicle dynamics and control*. Springer Science & Business Media.

[3] Jazayeri, S.A., Rad, M.S. and Azadi, S., 2005, January. Development and validation for mean value engine models. In *ASME 2005 Internal Combustion Engine Division Fall Technical*

[DOI: 10.22068/ijae.8.3.2792]

Conference (pp. 19-28). American Society of Mechanical Engineers.

[4] Nikzadfar, K. and Shamekhi, A.H., 2015. An extended mean value model (EMVM) for control-oriented modeling of diesel engines transient performance and emissions. *Fuel*, *154*, pp.275-292.

[5] Gaing, Z.L., 2004. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE transactions on energy conversion*, *19*(2), pp.384-391.

[6] Sedighizadeh, A.K.H.E.M., and Pirayesh, A.R.A., 2011, Optimal PID Controller Design for AVR System Using New optimization Algorithm. *International Journal on "Technical and Physical Problems of Engineering" (IJTPE), ISSN*, pp.2077-3528.

[7] Ou, C. and Lin, W., 2006, June. Comparison between PSO and GA for parameters optimization of PID controller. In *2006 International Conference on Mechatronics and Automation*.

[8] Mozaffari, A., Azad, N.L., Hedrick, J.K. and Taghavipour, A., 2016. A hybrid switching predictive controller with proportional integral derivative gains and GMDH neural representation of automotive engines for cold start emission reductions. *Engineering Applications of Artificial Intelligence*, *48*, pp.72-94.

[9] Abachizadeh, M., Yazdi, M.R.H. and Yousefi-Koma, A., 2010, July. Optimal tuning of PID controllers using artificial bee colony algorithm. In *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (pp. 379-384). IEEE.

[10] Lin, C.L., Jan, H.Y. and Shieh, N.C., 2003. GA-based multiobjective PID control for a linear brushless DC motor. *IEEE/ASME transactions on mechatronics*, *8*(1), pp.56-65.

[11] Bagis, A., 2007. Determination of the PID controller parameters by modified genetic algorithm for improved performance. *Journal of Information Science and Engineering*, *23*(5), pp.1469-1480.

[12] Bassi, S.J., Mishra, M.K. and Omizegba, E.E., 2011. Automatic tuning of proportional-integral-derivative (PID) controller using particle swarm optimization (PSO) algorithm. *International Journal of Artificial Intelligence & Applications*, *2*(4), p.25.

[13] Hendricks, E. and Sorenson, S.C., 1991. *SI engine controls and mean value engine modelling* (No. 910258). SAE Technical paper.

[14] Hendricks, E. and Luther, J.B., 2001, September. Model and observer based control of internal combustion engines. In *Proceedings of International Workshop on Modeling, Emissions and Control in Automotive Engines (MECA01)*.

[15] Hendricks, E., Jensen, M., Kaidantzis, P., Rasmussen, P. and Vesterholm, T., 1993. *Transient A/F ratio errors in conventional SI engine controllers* (No. 930856). SAE Technical Paper.

[16] Stefanopoulou, A.G., Cook, J.A., Grizzle, J.W. and Freudenberg, J.S., 1998. Control-oriented model of a dual equal variable cam timing spark ignition engine. *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF DYNAMIC SYSTEMS MEASUREMENT AND CONTROL*, *120*, pp.257-266.

[17] Gerhardt, J., Hönninger, H. and Bischof, H., 1998. *A new approach to functional and software structure for engine management systems-BOSCH ME7* (No. 980801). SAE technical paper.

[18] Heintz, N., Mews, M., Stier, G., Beaumont, A.J. and Noble, A.D., 2001. *An approach to*

[DOI: 10.22068/ijae.8.3.2792]

torque-based engine management systems (No. 2001-01-0269). SAE Technical Paper.

[19] Fiaschetti, J. and Narasimhamurthi, N., 1995. *A descriptive bibliography of SI engine modeling and control* (No. 950986). SAE Technical Paper.

[20] Arsie, I., Pianese, C., Rizzo, G., Flora, R. and Serra, G., 2000. *A computer code for SI engine control and powertrain simulation* (No. 2000-01-0938). SAE Technical Paper.

[21] Caraceni, A., De Cristofaro, F., Ferrara, F., Scala, S. and Philipp, O., 2003. *Benefits of using a real-time engine model during engine ECU development* (No. 2003-01-1049). SAE Technical Paper.

[22] Hendricks, E. and Sorenson, S.C., 1990. *Mean value modelling of spark ignition engines* (No. 900616). SAE Technical paper.

[23] Chevalier, A., Müller, M. and Hendricks, E., 2000. *On the validity of mean value engine models during transient operation* (No. 2000-01-1261). SAE Technical Paper.

[24] Hendricks, E., Chevalier, A., Jensen, M., Sorenson, S.C., Trumpy, D. and Asik, J., 1996. *Modelling of the intake manifold filling dynamics* (No. 960037). SAE Technical Paper.

[25] Heywood, J.B., 1988. *Internal combustion engine fundamentals* (Vol. 930). New York: Mcgraw-hill.

[26] Moskwa, J.J., 1988. *Automotive engine modeling for real time control* (Doctoral dissertation, Massachusetts Institute of Technology).

[27] Weeks, R.W. and Moskwa, J.J., 1995. *Automotive engine modeling for real-time control using matlab/simulink* (No. 950417). SAE Technical Paper.

[28] Moskwa, J.J. and Hedrick, J.K., 1992. Modeling and validation of automotive engines for control algorithm development. *Journal of dynamic systems, measurement, and control*, *114*(2), pp.278-285.

[29] Sandoval, D., 2002. *An improved friction model for spark ignition engines* (Doctoral dissertation, Massachusetts Institute of Technology).

[30] Holland, J.H., 1975. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.

[31] Kim, H.S. and Cho, S.B., 2000. Application of interactive genetic algorithm to fashion design. *Engineering applications of artificial intelligence*, *13*(6), pp.635-644.

[32] Simon, D., 2013. *Evolutionary optimization algorithms*. John Wiley & Sons.

[33] Mozaffari, A., Gorji-Bandpy, M. and Gorji, T.B., 2012. Optimal design of constraint engineering systems: application of mutable smart bee algorithm. *International Journal of Bio-Inspired Computation*, *4*(3), pp.167-180.

[34] Rao, S.S. and Rao, S.S., 2009. *Engineering optimization: theory and practice*. John Wiley & Sons.

[35] Eberhart, R.C. and Kennedy, J., 1995, October. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science* (Vol. 1, pp. 39-43).

[36] Binitha, S. and Sathya, S.S., 2012. A survey of bio-inspired optimization algorithms. *International Journal of Soft Computing and Engineering*, *2*(2), pp.137-151.

[37] Shi, Y. and Eberhart, R., 1998, May. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational*

[DOI: 10.22068/ijae.8.3.2792]

Intelligence., The 1998 IEEE International Conference on (pp. 69-73). IEEE.

[38] Clerc, M. and Kennedy, J., 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, *6*(1), pp.58-73.

[39] Mehrabian, A.R., and Lucas, C., 2006. A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, *1*(4), pp.355-366.

[40] Guzzella, L., and Onder, C., 2009. *Introduction to modeling and control of internal combustion engine systems*. Springer Science & Business Media.

[41] Karaboga, D. and Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, *39*(3), pp.459-471.

[42] Honek, M., Wojnar, S., Simoncic, P. and Rohar-Hkiv, B., 2010. Control of electronic throttle valve position of SI engine. In *International Conference February* (Vol. 10, p)

[DOI: 10.22068/ijae.8.3.2792]