Prismatic Series Elastic Actuator: Modelling and Control by ICA and PSO-Tuned Fractional Order PID

S. Norouzi Ghazbi* & A. R. Akbarzadeh
Department of Mechanical Engineering,
Ferdowsi University of Mashhad, Iran
Center of Excellence on Soft Computing and Intelligent Information Processing (SCIP), Ferdowsi university of Mashhad, Iran
E-mail: somaye_noroozi@yahoo.com, ali_akbarzadeh_t@yahoo.com
*Corresponding author

Received: 8 April 2015, Revised: 20 July 2015, Accepted: 22 September 2015

Abstract: This paper presents dynamic modelling and control of a linear prismatic series elastic actuator. Since this actuator has the capability of generating large torques, it is increasingly used in the human-assistive robotic systems. Due to having the human in the loop, the actuator requires precise control. A fractional PID controller is used for the control to improve performance because this controller has more additional degrees of freedom than the classical PID. The actuator has one servo driver and five controller gains to be tuned. The gains are optimized using both Particle Swarm Optimization (PSO) and Imperialist Competitive Algorithms (ICA). Comparison of the results from the two optimization methods illustrates that the PSO tuned FOPID controller has a slightly better performance, faster convergence and better settling time. Next, the PSO tuned controller is compared with a Genetic Algorithm (GA) tuned PID controller. It is shown that the PSO tuned FOPID controller continues to offer better performance, especially in terms of rise time and settling time.

Keywords: Control, Fractional Order PID, Imperialist Competitive Algorithm, Particle Swarm Optimization, Series Elastic Actuator;

Reference: Norouzi Ghazbi, S., Akbarzadeh, A. R., "Prismatic Series Elastic Actuator: Modelling and Control by ICA and PSO-Tuned Fractional Order PID", Int J of Advanced Design and Manufacturing Technology, Vol. 8/ No. 4, 2015, pp. 23-32.

Biographical notes: A. Akbarzadeh received his PhD in Mechanical Engineering in 1997 from the University of New Mexico in USA. He worked at Motorola, USA, for 15 years where he led R&D as well as automation teams. He joined the Ferdowsi University of Mashhad in 2005 and is currently a full professor in the Mechanical Engineering Department. He has over 45 journal publications and over 60 conference papers. His areas of research include robotics (parallel robots, biologically inspired robots, bipedal robots, and rehabilitation robotics), dynamics, kinematics, control, automation, optimization as well as design and analysis of experiments. He is also a founding member of the Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP). **S. Norouzi** G. received her MS degree in Mechanical engineering from Ferdowsi university of Mashhad 2013. Her current research interests are robust control method, system identification and optimization.

1 INTRODUCTION

A series elastic actuator (SEA), shown in Fig. 1 is a variable stiffness actuator and includes an elastic element, a linear spring, placed between the gear motor and the load. An additional elastic element is used to elastically decouple the actuator from the load and improves tolerance to mechanical shocks. If proper stiffness is selected based on the desired task, then the spring can provide additional protection of the motor and gearbox in the case of unwanted collisions on the output links [1]. The SEAs are increasingly applied to human-assisted robotic systems because of this ability. Human assistive robots, i.e., systems that assist human motions with actuation capabilities, have been intensively developed in recent years [1-2]. Actuators with capability of generating large torques are required to effectively assist human motions. In human-assistance scenarios, the actuator is connected to single or multiple joints of human in order to provide assistive torques. Due to having a patient in the control loop, meeting precise control demands is important and required. The increasingly usage of series elastic actuators in human assistive robots and the need for precise control has drawn a lot of attention in recent years. Control of SEA with a PID-type controller was proposed by Au [3] in 2006. In 2009 Kong [4] presented force control of a rotary series elastic actuator. A study on force control of a SEA with rotary spring by using PID controller was reported by Taylor [5] in 2011. Hutter [6] in 2011 studied fast position control of high compliant series elastic actuator. Misgeld et al., [7] in 2014 studied robust control of adjustable compliant actuators.

Fig. 1 Series elastic actuator

The Fractional Order Calculus (FOC) constitutes a branch of mathematics that deals with derivatives and integrals from non-integer orders. Although FOC is a more than 300 years old topic, its great consequences in contemporary theoretical and experimental research including in the area of control theory have been widely discussed recently. Fractional order proportional-integral-derivative (FOPID) controllers have received a considerable attention in the last years both from academic and industrial point of view. As there are five parameters to select instead of three in standard PID type controllers, these controllers provide more flexibility in the controller design. However, this also implies that tuning of the controller can be much more complex. During last years, several techniques have been suggested for tuning of the controller gains.

For the first time, the concept of FOPID controller and its better performance in comparison with the classical PID controller was introduced by Podlunbny [8] in 1997.

Vinagre [9] in 2000, studied a frequency domain approach using FOPID controllers. An optimization method where the parameters of the FOPID are tuned with predefined methods is presented in 2004 by Monje et al., [10]. In 2006, Valerio and Costa [11] proposed Zigler-Nichols tuning rules for FOPID. In 2009, an optimization method with particle swarm optimization was reported by Zamani et al., [12]. A tuning and auto tuning method for FOPID controllers based on gain crossover frequency and phase margin was proposed in 2007 by Monje et al., [13]. Biswas et al., [14] in 2009 studied gain tuning of FOPID with an improved Differential Evolution (DE) algorithm. Another DE-tuned FOPID controller was reported by Shahri et al., [15] in 2014. Tuning of Fractional order PID controller by using Covariance Matrix Adaptation Evolution Strategy (CMAES) algorithm is presented in 2014 by Sivananaithaperumal and Baskar [16].

Imperialist Competitive Algorithm (ICA) is a novel evolutionary optimization algorithm. ICA was introduced firstly in 2007 By Atashpaz-Gargari and Lucas [17]. This method has extensively been used to solve different kinds of optimization problems. In [12], [13] and [18] ICA is used to optimize PID type controller's parameters for SISO systems.

Particle Swarm optimization (PSO) is one of the most important swarm intelligence paradigms [19].In 2012 a study on comparison of two optimization methods of gain tuning of FOPID controller: differential evolution and PSO algorithm, was reported by Bingul and Karahan [20]. Mohamed Hussain et al., [21] in 2014 studied PID controller tuning methods in comparison with PSO algorithm. Saleem and Taha [22] in 2015 used PSO-tuned cascade PID controller to control a servo-pneumatic system.

This paper presents dynamic modelling and control of a prismatic series elastic actuator. Fractional order PID controller is selected as the control rule. In order to optimize the controller gains, two optimization methods: PSO algorithm and ICA are used and compared.

The outline of this paper is as follows. Dynamic modelling of the system is presented in section 3 and a FOPID controller is designed in section 4. The procedure of gain optimization and results are presented in section 5 and 6, respectively. Finally, concluding discussion is given in section 7.

2 MODELLING

A simple model for an actuator with a series elastic element is shown in Fig. 2.

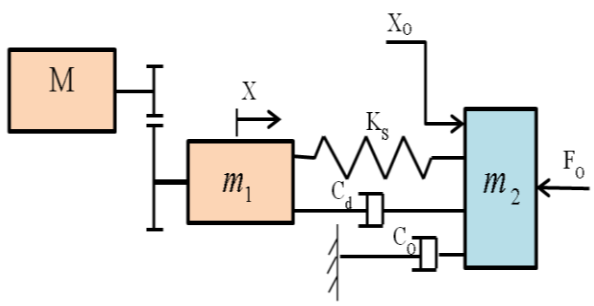

Fig. 2 A series Elastic Actuator model

The notations used in Fig. 2, represent: motor: *M*, mass of nut and ball screw: m_1, spring stiffness: K_s, coulomb friction constant of guides: C_d, coulomb friction constant between load and fixed part of SEA: C_o, load mass: m_2, movement of load: X_o, movement of shaft: *X* and output force: F_o. The friction is assumed to be negligible [23], and then control of a simple model is proposed.

Referring to Fig. 2, by applying Newton's law the relations between X_o, motor force, F_m, and F_o can be written as below (Eq. 1 and 2),

$$m_1\ddot{x} = F_m - k_s(x - x_o) \tag{1}$$

$$F_o = k_s(x - x_o) \tag{2}$$

By taking Laplace transform, an expression relating F_m and F_o can be found as,

$$F_o(s) = \left(\frac{k_s}{m_1 s^2 + k_s}\right)F_m(s) - \left(\frac{k_s m_1 s^2}{m_1 s^2 + k_s}\right)X_o(s) \tag{3}$$

Equation 3 shows how F_m needs to vary to give a desired output force while the output load is moving. The block diagram of the system is shown in Fig. 3.

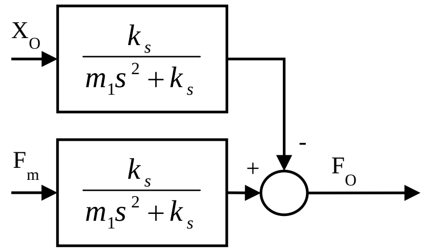

Fig. 3 The block diagram of the system

3 CONTROL

Figure 4 illustrates a drawing of the proposed control scheme. It includes the feed forward terms and the FOPID control loop.

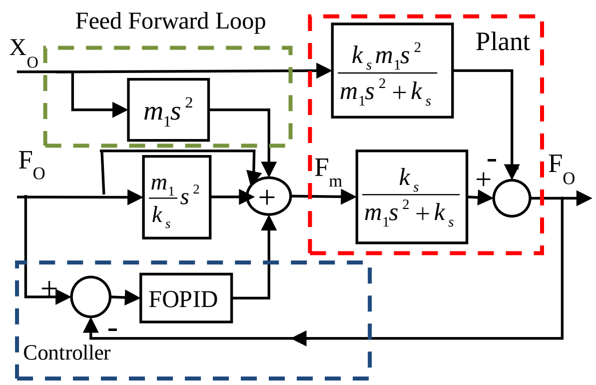

Fig. 4 The proposed control system

The feed forward gains can be obtained by,

$$F_m(s) = F_o(s)\left(\frac{m_1}{k_s}s^2 + 1\right) + m_1 s^2 X_o(s) \tag{4}$$

The value of the parameters in the above transfer functions are listed in the table 1.

Table 1 Transitions selected for thermometry

Parameter	k_s (N/m)	m_1 (kg)
value	60	0.7

A. Fractional Calculus

The fundamental operator ${}_aD_t^r f(t)$, is a combined differentiation-integration operator commonly used in fractional calculus. The ${}_aD_t^r$ is called the fractional derivative-integral of order r with respect to variable t and with the starting point a. It is defined as Eq. (5) [5]:

$$
{}_aD_t^r = \begin{cases} \dfrac{d^r}{dt^r} & r > 0, \\ 1 & r = 0, \\ \int_a^t (d\tau)^{-r} & r < 0, \end{cases} \tag{5}
$$

The most frequently used three definitions for the fractional derivatives are the Grunwald-Letnikov (GL), the Riemen-Liouville (RL) and the Caputo definition [24]. The GL fractional derivative which is used here has the following form (Eq. (6)),

$$
{}_aD_t^r f(t) = \lim_{h\to 0} h^{-r} \sum_{j=0}^{[\frac{t-a}{h}]} (-1)^j \binom{r}{j} f(t-jh)[0,1] \tag{6}
$$

Where [.] designates the integer part.
For the computation of the coefficients to obtain the numerical solution, when α has a fixed value of derivative order, the following recursive formula (Eq. (7) and (8)) can be used [25]:

$$
{}_aD_t^r f(t) \approx \lim_{h\to 0} h^{-r} \sum_{j=0}^{[\frac{t-a}{h}]} \omega_j^{(\alpha)} f(t-jh) \tag{7}
$$

Where,

$$
\omega_0^{(\alpha)} = 1, \quad \omega_j^{(\alpha)} = \left(1 - \frac{\alpha+1}{j}\right)\omega_{j-1}^{(\alpha)} \tag{8}
$$

Laplace transform of non-integer order derivatives is a fundamental tool in design of a FOPID controller. In the case of Laplace transform, there is no significant difference with respect to the classical case. Besides, Inverse Laplace transformation is necessary for time domain representation of the system. The Laplace transformation is described by the following equation (Eq. 9) [20]:

$$
L[{}_aD_t^\alpha f(t)] = s^\alpha L[f(t)] \tag{9}
$$

B. Fractional order PID (FOPID) Controller

The integral-differential equation defining the control action of a fractional order PID ($PI^{-\lambda}D^{\mu}$) controller is given by (Eq. 10):

$$
u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) \tag{10}
$$

Where $e(t)$ is the error signal of the tracking system, $u(t)$ is the control signal, λ and δ are positive real numbers; K_p is the proportional gain K_i is the integration constant and K_d is the differentiation constant. The $PI^{-\lambda}D^{\mu}$ Controller, also known as $PI^{\lambda}D^{\delta}$ controller, was studied in time domain in [26] and in frequency domain in [27].
The initial values for the controller's parameters were found with the traditional Ziegler-Nichols (Z-N) method [28]. Sustained oscillations are observed quite easily and the Z-N formula gives initial values for proportional, integrative and derivative parameters as shown in table 2.

Table 2 The initial values for Z-N tuned FOPID controller

Parameter	K_p	K_i	λ	K_d	μ
value	2	0	-	3	0.26

The time response of the controlled system which is tuned by Z-N tuning technique is shown in Fig. 5.

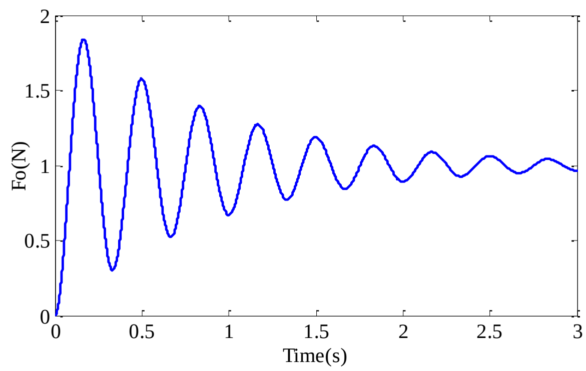

Fig. 5 Closed loop response for the Z-N tuned FOPID controller

Characteristics of the response, settling time and overshoot are 3.08s and 86%, respectively.

4 OPTIMIZATION METHODS

Evolutionary algorithms differ from the traditional optimization techniques in that EAs make use of a population of solutions, not a single point solution. Thus, EAs can find multiple optimal solutions in one single run due to their population-approach. Considering many points in the search space, it is more likely converging to the global optimum. An iteration of EA involves a competitive selection that weeds out poor solutions and offspring generation mechanism. Several evolutionary search algorithms like GA, DE, PSO, ICA were developed independently. These algorithms differ in selection, offspring generation and replacement mechanisms. In order to solve global functional optimization problems, PSO is a well-known algorithm and ICA is a more recently introduced optimization method , hence these algorithms are employed in this paper. These algorithms are briefly explained in this section.

A. The PSO algorithm

PSO is one of the modern Heuristics algorithms and was developed through simulation of a simplified social system. It has been found that this optimization algorithm is robust in solving continuous non-linear optimization problems [29].
This method is based on evolutionary computation technique. The basic PSO is developed from research on the social behavior of the birds and fishes. Individual birds and fishes exchange information about their position, velocity and fitness, and the behavior of the flock is then influenced to increase the probability of migration to regions of high fitness [29-30]. In PSO, instead of using genetic operators, individuals called as particles are "evolved" by cooperation and competition among themselves through generations [31].
A particle represents a potential solution to a problem. Each particle adjusts its flying according to its own flying experience and its companion flying experience. Each particle is treated as a point in a D-dimensional space. The i th particle is represented as:

$$X_I = \left(x_{i1},\ x_{i2}, \ldots, x_{in}\right)$$

The best previous position (giving the minimum fitness value) of any particle is recorded and represented as $P_I = \left(p_{i1},\ p_{i2}, \ldots, p_{iD}\right)$, this is called *pbest*. The index of the best particle among all particles in the population is represented by the symbol *g*, called as *gbest.* The velocity of the particle i is represented as:

$$V_i = \left(v_{i1},\ v_{i2},\ \ldots,\ v_{iD}\right).$$

The particles are updated according to the following equations (Eq. 11 & 12):

$$V_{i,m}^{(t+1)} = W.V_{i,m}^{(t)} + C_1 \times rand\,() \times (Pbest - X_{i,m}^{(t)}) + C_2 \times rand\,() \times (gbest - X_{i,m}^{(t)}) \tag{11}$$

$$X_{i,m}^{(t+1)} = X_{i,m}^{(t)} + V_{i,m}^{(t+1)} \tag{12}$$

Where C_1 and C_2 are two positive constant. As recommended in [31], the constants are $C_1=C_2=1.49$. While rand () is random function between 0 and 1, and n represents iteration. Eq.11 is used to calculate particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group's best experience. Then the particle flies toward a new position.
Kumar and Gupta reported in [32] that limiting the particle velocity in the *i*-th dimension by some maximum and minimum value (Eq. 13 & 14) enhances the local exploration of the problem space.

$$if \quad v_{i,m}^{(t+1)} > V_m^{\max}, then \quad v_{j,m}^{(t+1)} = V_m^{\max} \tag{13}$$

$$if \quad v_{i,g}^{(t+1)} < V_g^{\min}, then \quad v_{j,g}^{(t+1)} = V_g^{\min} \tag{14}$$

The performance of each particle is measured according to a predefined fitness function which is related to the problem to be solved. Inertia weight, w , is brought into the equation to balance between the global search and local search capability. It can be a positive constant or even positive linear or nonlinear function of time.
The pseudo-code of the algorithm is as follows:

1. Select some random points on the function and initialize the individuals of the population.
2. Calculate each individual's Evaluation Value (EV).
3. Compare each individual's EV with its Pbest and select the individual with the best EV as gbest.
4. Modify the member velocity v of each individual.
5. Modify the member position of each indivual.
6. If change of each individual's velocity value is less than the criteria value stop, if not go to 2.
7. The individual that generates the latest gbest is an optimal controller parameter.

B. The ICA algorithm

Imperialist Competitive Algorithm is a global search strategy that uses the socio-political competition among empires as a source of inspiration.
Like other evolutionary ones that start with initial population, ICA begins with initial empires. Any individual of an empire is a country. 'Country' is an array of variables (p_i) to be optimized. It is similar to 'chromosome' in GA terminology. This array is defined by $Country^* = [p_1, p_2, p_3, ..., p_{N_{var}}]$. In an N_{var}-dimensional optimization problem, a country is a $1 \times N_{var}$ array. There are two types of countries; colony and imperialist state that collectively form empires. Imperialistic competitions among these empires form the basis of the ICA [33].
Some of the best countries (in optimization terminology, countries with the least cost) are selected

to be the imperialist states and the rest form the colonies of these imperialists. All the colonies of initial countries are divided among the mentioned imperialists based on their power. To proportionally divide the colonies among imperialists, the normalized cost of an imperialist ($Nor.C_n$) is defined by:

$$Nor.C_n = c_n - \max\{c_i\} \tag{15}$$

Where C_n and $Nor.C_n$ are respectively the cost of the n-th imperialist and its normalized cost. Then, the normalized power of each imperialist is defined by (Eq. 16):

$$P_n = \left| \frac{Nor.C_n}{\sum_{i=1}^{N_{imp}} Nor.C_i} \right| \tag{16}$$

Where N_{imp} is the number of imperialists. Then the initial number of colonies of an empire can be calculated by (Eq. 17):

$$Num.Col_n = round\{p_n.N_{col}\} \tag{17}$$

Where $Num.Col_n$ is the initial number of colonies of the n-th empire and N_{col} is the number of all colonies. These colonies along with the imperialist will form n-th empire.

After forming initial empires, the colonies in each of them start moving toward their relevant imperialist country. In this movement each colony moves toward the imperialist by x units and a deviation of θ from the direct path between the colony and the imperialist. x is a random variable with uniform (or any proper) distribution. Then (Eq.18):

$$x \sim U(0, \beta \times d) \tag{18}$$

Where β is a number greater than one and d is the distance between the colony and the imperialist state. $\beta>1$ causes the colonies to get closer from both sides. Also θ is a parameter with uniform (or any proper) distribution. Then (Eq.19):

$$\theta \sim U(-\gamma, \gamma) \tag{19}$$

Where γ is a parameter that adjusts the deviation from the original direction. Deviation of θ is added to the direction of movement to increase the ability of searching more area around the imperialist. By adding this deviation, assimilating the colonies by the imperialist states would not happen in direct movement of the colonies toward the imperialist. In most implementations, a value of about two for β and about $\pi/4$ (rad.) for γ results in good convergence of countries to the global minimum [34].

During this competition, weak empires collapse and powerful ones take possession of their colonies. In modelling collapse mechanism different criteria can be defined for considering an empire powerless. In most of the implementations, it is assumed that an empire would be collapsed and eliminated, when it loses all of its colonies. If after this movement one of the colonies possesses more power than its relevant imperialist, they will exchange their positions. To begin the competition between empires, total objective function of each empire should be calculated.

The total power of an empire depends on both the power of the imperialist country and the power of its colonies. This fact is modelled by defining the total power of an empire as the power of imperialist country plus a percentage of mean power of its colonies (Eq. 20),

$$\begin{aligned} T.PC_i = {} & PC(imprialist_i) + \\ & \zeta\, mean\{PC(Colonies\ \ of\ \ empire_i)\} \end{aligned} \tag{20}$$

Where $T.PC_i$ is the total cost of the i-th empire and ζ is a positive small number. A little value for ζ causes the total power of the empire to be determined by just the imperialist and increasing it will increase the role of the colonies in determining the total power of an empire. The value of 0.1 for ζ has shown good results in most of the implementations [35].

Imperialistic competition converge to a state in which there exist only one empire and its colonies are in the same position and have the same cost as the imperialist. The pseudo-code of the algorithm is as follows [34].

1. Select some random points on the function and initialize the empires.
2. Move the colonies toward their relevant imperialist (assimilation).
3. If there is a colony in an empire which has lower cost than that of the imperialist, exchange the positions of that colony and the imperialist.
4. Compute the total cost of all empires (related to the power of both the imperialist and its colonies).
5. Pick the weakest colony (colonies) from the weakest empires and give it (them) to the empire that has the most likelihood to possess it (imperialistic competition).
6. Eliminate the powerless empires.
7. If there is only one empire left, stop, if not go to 2

5 RESULTS

In the both methods, the number of selected generation is 100 and the population size is 50 , the other PSO and ICA specific parameters are listed in table 3 and 4, respectively.

Table 3 The Selected PSO parameters

Parameter	Value
Inertia weight factor	0.4
Acceleration constant C_1	0.82
Acceleration constant C_2	0.82

Table 4 The Selected ICA parameters

Parameter	Value
Revolution rate	0.3
Number of countries	100
Number of initial Imperialists	5
Number of Decades	100
Assimilation coefficient	0.5
Assimilation angle coefficient	0.5
Zeta	0.02
Parameter	Value

For optimal tuning of the controller parameters using evolutionary algorithms, it is necessary to use a proper objective function. In this paper, the transient and steady state response of the system are used to evaluate the performance of the designed FOPID controller. The characteristics representing transient and steady state response- Overshoot (M_p), Settling time (T_s) and integral of time weighted absolute error ($ITAE$)- are used to evaluate the designed controller. A good controller results in the output to have low values for T_r , M_p and T_S . The multi objective design problem is converted to single objective one by considering a linear combination of all criteria. Therefore (Eq.21)

$$\text{Cost function}=w_1.ITAE + w_2.M_P + w_3.T_s \qquad (21)$$

W_i s are the weights which must be determined by designer.

The optimization problem can be stated as minimizing cost function considering the following constraints:

Table 5 Constraints

Parameter	Value
Variable min (K_P , K_d ,Miu)	(0,0,0.001)
Variable min (K_P , K_d ,Miu)	(15,15,1)

Figure 6 shows the initial population of each empire for ICA algorithm. In this figure the first imperialist has formed the most powerful empire and has the greatest number of colonies.

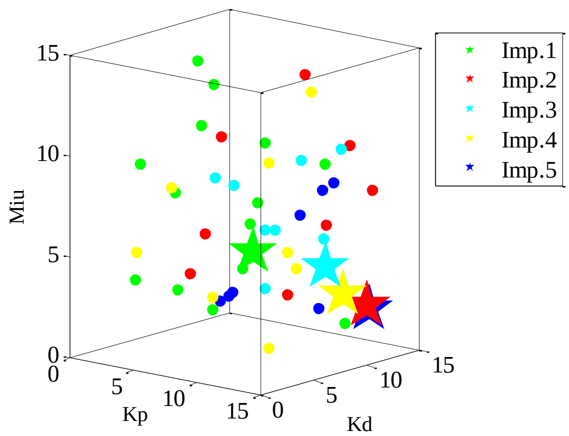

Fig. 6 Initial Empires

Empires at iteration (decade) 50 and 100 are shown in Fig. 7 and Fig. 8.

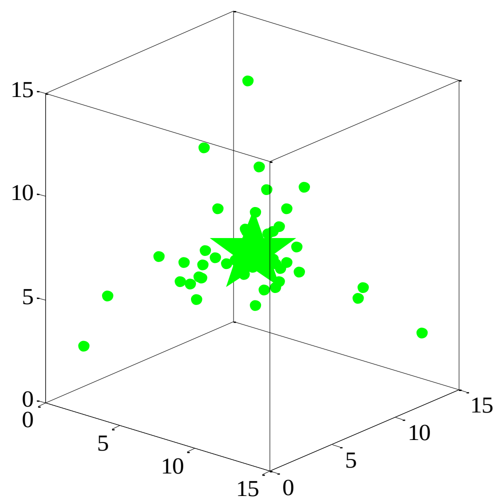

Fig. 7 Empires at Iteration 50

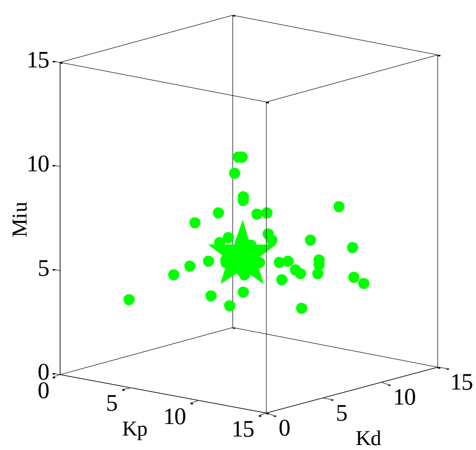

Fig. 8 Empires at Iteration 100

Both ICA and PSO algorithms are programmed in MATLAB and run on an ADM PC, CPU 65X2 Dual Core Processor 4200 2.2 GHz, RAM 1GB.
By minimizing the objective function, the optimal values of parameters and computation time are as below (table 6):

Table 6 The optimal values of parameters and computation

Optimal values	PSO	ICA
K_P	1.073	1.44
K_d	15	14.9
Miu	0.208	0.24
MP	67%	68%
T_s	0.97	1.04
Cost function	1019	1046
Computation time (hr)	3	10.5

By using optimization method, 20% and 66% improvement in settling time and overshoot, are achieved respectively.
Figure 9 shows the convergence history for the best cost and mean cost of ICA, and Fig. 10 demonstrates the iteration process of cost for ICA and PSO methods. In Fig. 9 Mean cost is average of all empires' cost. As it is observed after the number of empires reduced to 1 in decade 43 mean cost is calculated based on only the remained empire and is equal to it.

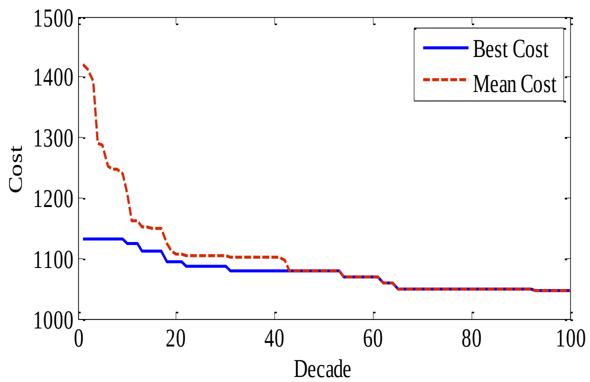

Fig. 9 Best and Mean cost of ICA method

In Fig. 10 a significant decrease in the target function is seen in the beginning of the evolution process. After about 22 and 60 iterations for PSO algorithm and ICA respectively, the changes in the fitness function become relatively minute.

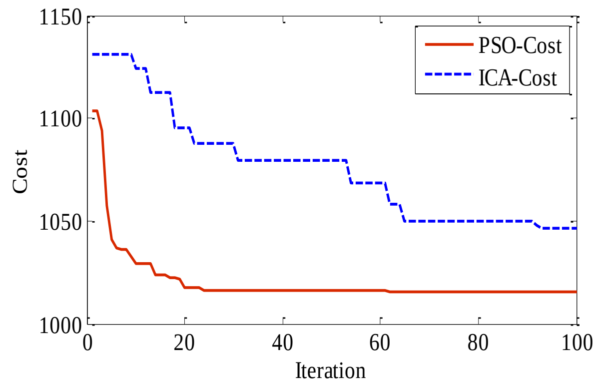

Fig. 10 Cost of ICA method

As it can be observed in Fig. 9 the PSO algorithm produces the least cost and performs better than the other algorithm. Closed loop response of SEA system with PSO-tuned and ICA-tuned FOPID are illustrated in Fig. 11.

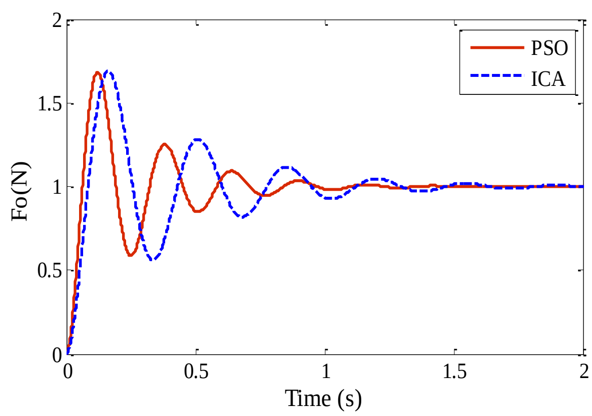

Fig. 11 Cost of ICA method

Considering Fig. 11, it can be resulted that there is no significant difference between the performance of ICA and PSO algorithm in tuning of FOPID controller.
To evaluate the performance of FOPID controller, it was compared with the GA-tuned PID controller. Genetic Algorithm is selected as the optimization method because many researches confirmed that GA has a good performance in tuning of PID controller [36]. The closed loop responses of SEA system to a unit step in the presence of PID and FOPID controllers are given Fig. 12. It is worth mentioning that optimized value of the objective function for the SEA system with GA-PID controllers is 3720. Also settling time and overshoot are 1.52 (s) and 67% respectively.

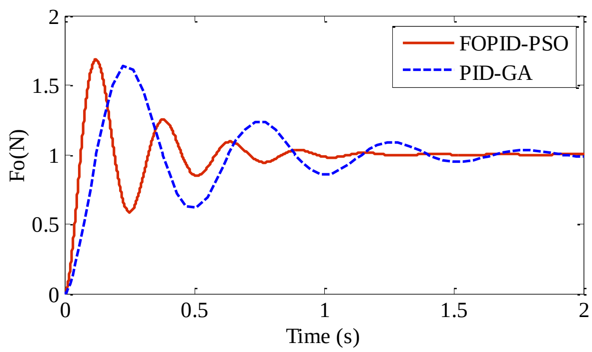

Fig. 12 Cost of ICA method

It can be observed from the Fig. 12 that the FOPID controller gives better performance with respect to the regular PID controller, especially in terms of settling time.

6 CONCLUSION

In this paper, dynamic modelling and control of a series elastic actuator with a fractional order PID controller was proposed. Optimization of the controller's gains was carried out using two optimization method: PSO algorithm and ICA method. Comparison of the results with the one obtained from the Zigler-Nichols tuned FOPID controller showed 20% and 66% improvement in overshoot and settling time, respectively. Next, performance of the two optimization methods was compared. Results showed that the performance of the two tuned controllers did not differ significantly; about 7% and 2% improvement in settling time and overshoot is achieved using PSO algorithm. However, the average computation time used to tune the FOPID by ICA method was about 2 times more time consuming than PSO algorithm. Finally, to evaluate the performance of the optimized FOPID, a comparison between the PSO-tuned FOPID and GA-PID was made. Results show that the optimized PSO-tuned FOPID has about 50% faster response time than the GA-tuned PID.

REFERENCES

[1] Accoto, A, Carpino G, Sergi F, et al., “Design and Characterization of a Novel High Power Series Elastic Actuator for a Lower Limb Roboic Orthosis”, International Journal of Advanced Robotic Systems, Vol. 10, No. 359, 2013, pp. 1-12.

[2] Carpino, G., Accoto, D., Sergi, F., Luigi., T., N., and Guglielmelli, E., “A Novel Compact Torsional Spring for Series Elastic Actuators for Assistive Wearable Robots”, Journal of Mechanical Design, Vol., 134 , No. 12, 2012.

[3] Au, S., k., Dilworth, P., and Herr, H., “An Ankle-Foot Emulation System for the Study of Human Walking Biomechanics”, International Conference on Robotics and Automation, Orlando, Florida, May, 2006.

[4] Kong, K., Bae, J., and Tomizuka, M., “Control of rotary series elastic actuator for ideal force-mode actuation in human-robot interaction applications”, IEEE/ASME Transactions on Mechatronics, Vol. 14, No. 1, 2009.

[5] Taylor, D., M., “A Compact Series Elastic Actuator for Bipedals with Human-like Dynamic Performance”, MSc Thesis, Crnegie Mellon University, 2011.

[6] Hutter, M., Remy, C. D., Hoepflinger, M. A., and Siegwart, R., “High Compliant Series Elastic Actuation for the Robotic Leg Scar IETH”, International Conference on Climbing and Walking Robots (CLAWAR), Paris, France, September 7, 2011.

[7] Misgeld, B., J., E., Geriach-Hahn, K., Ruschen, D., Popprapa, A., and Leonhardt, S., “Control of Adjustable Compliant Actuators”, Machines, Vol. 2, No. 2, 2014, pp. 134-157.

[8] Podlubny, I., Dorcak, L., and Kostial, I., “On Fractional Derivatives, Fractional-Order Dynamic Systems and PID Controllers”, Proceedings of the 36th Conference on Decision & Control, San Diego, California, USA, 1997.

[9] Vinagre, B. M., Podlubny, I., Dorcak, L., and Feliu, V., “On Fractional PID Controllers: A Frequency Domain Approach”, Proceedings of IFAC Workshop on Digital Control: Past, Present and Future of PID Control. Terrasa, Spain, 2000, pp. 53-58.

[10] Monje, C. A., Vinagre, B. M. , Chen, Y. Q., Feliu, V., Lanusse, P., and Sabatier, J., “Proposals for Fractional Tuning”, Proceedings of Fractional Differentiation and its Applications, Bordeaux, 2004.

[11] Vale´rio, D., Costa, J. S., “Tuning of Fractional PID Controllers with Ziegler–Nichols- Type Rules, Signal Processing”, Signal Processing , Vol. 86, 2006, pp. 2771–2784.

[12] Zamani, M., Karimi,G. M., Sadati, N., and Parniani, M., “Design of a Fractional Order PID Controller for an AVR using Particle Swarm Optimization”, Control Engineering practice, Vol. 17, No. 12, 2009.

[13] Monje, A., Blas, M., Vinagre, M., Feliu, V., and Chen, Y., “Tuning and Auto-Tuning of Fractional Order Controllers for Industry Applications”, Control engineering practice, Vol. 16, No. 7, 2008.

[14] Biswas, A., Das, S., Abraham, A., and Dasgupta, S., “Design of Fractional Order PID Controllers with an Improved Differential Evolution”, Engineering Applications of Artificial Intelligence, Vol. 22, 2009, pp. 343-350.

[15] Shahri, M., Balochian, S., Balochian, H., and Zhang, Y., “Design of Fractional-order PID Controllers for Time Delay Systems Using Differential Evolution Algorithm”, Indian Journal of Science and Technology, Vol. 7, Issue 9, 2014.

[16] Sivananaitha perumal, S., Baskar, S., “Design of Multivariable Fractional Order PID Controller

using Covariance Matrix adaptation Evolution Strategy", Archives of control sciences, Vol. 24, No. 2, 2014.
[17] Atashpaz-Gargari E, Lucas C., "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialist Competition", IEEE Congress on Evolutionary Computation, Singapore, Singapore, 2007, pp. 4661-4667.
[18] Ahmadi, M., Baojahmadi, M., Fallah, S. N., and Karami, K., "PI-Controller Adjustment for a Lab Scale CSTH" by using Imperialist Compettitive Algorithm", IEEE 3rd International Conference on System Engineering and Technology (ICSET), Shah Alam, 2013, pp. 168-172.
[19] Kecskés, I., Székacs, L., Fodor, J., C., and Odry, P., "PSO and GA Optimization Methods Comparison on Simulation Model of a Real Hexapod Robot", IEEE 9th International Conference on Computational Cybernetics, Tihany, Hungary, 2013.
[20] Bingul, Z., Karahan, O., "Fractional PID Controllers Tuned by Evolutionary Algorithms for Robot Trajectory Control", Turk J Elec & Comp Sci, Vol., 20, No. 1, 2012.
[21] Mohamed Hussain, K., Zepherin, R., A., R., Kumar, M., Sh., and Kumar, S., M., G., "PID Controller Tuning Methods Comparison with Particle Swarm Optimization for FOPTD System", International journal for innovative research in science and technology, Vol. 1, No. 4, 2014, pp. 88-93.
[22] Saleem, A., Taha, B., Tutunji, T., and Al-Qiasia, A., "Identification and Cascade Control of Servo-Penumatic System using Particle Swarm Optimization", Simulation modeling practice and theory, Vol. 52, pp. 164-179.
[23] Williamson, M. M., "Series Elastic Actuator", MSc Thesis", Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1995.
[24] Vinagre, B. M., Podlubny, I., Petras, I., and Chen, Y.Q., "Using Fractional Order Adjustment Rules and Fractional Order Reference Models in Model Reference Adaptive Control", Nonlinear Dynamics, Vol. 29, 2001, pp. 269-279.
[25] Monje, C. A, Chen, Y. Q. and Vinagre, B.M., Book: Fractional Order Systems and Controls Fundamentals and Applications, Chapter 2, springer, 2010.
[26] Podlubny, I., "Fractional-Order Systems and Controllers", IEEE Transactions on Automatic Control, Vol. 44, No. 1, 1999, pp. 208-214.
[27] Petr´aˇs, I., "The Fractional-Order Controllers: Methods for Their Synthesis and Application", Journal of Electrical Engineering, Vol. 50, No. 9-10, 1999, pp. 284-288.
[28] Valerio, D., Costa, J., S., "Tuning of Fractional PID Controllers with Ziegler-Nichols-Type Rules", signal processing, Vol. 86, No. 10, 2006, pp. 2771-2784.
[29] Zhao, J., Li, T., Qian, J., "Application of Particle Swarm Optimization Algorithm on Robust PID Controller Tuning. Advances in Natural Computation: Book Chapter", Springer Berlin Heidelberg, 2005, pp. 948-957.
[30] Iwan, S. M., Fook, T., L., and Leap, k. M., "Tuning of PID Controller Using Particle Swarm Optimization (PSO)", Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, 2011.
[31] Sultaniya, Sh., Gupta, R., "Design of PID Controller using PSO Algorithm for CSTR System", International Journal of Electronic and Electrical Engineering, Vol. 7, No. 9, 2014, pp. 971-977.
[32] Kumar, A., Gupta, R., "Compare the Results of Tuning of PID Controller by using PSO and GA Technique for AVR System", International Journal of Advanced Research in computer engineering and technology, Vol. 2, No. 6, 2013.
[33] Mohammadzadeh, A., Atashpaz, G., E., and Lucas, C., "Vehicle Fuzzy Controller Design Using Imperialist Competitive Algorithm", 2nd Iranian Joint Congress on Fuzzy and Intelligent Systems, Tehran, Iran, 2008.
[34] Khabbaz, A., Atashpaz-Gargari, E., Lucas, C., "Imperialist Competitive Algorithm for Minimum Bit Error Rate Beamforming", Int. J. Bio-Inspired Computation, Vol. 1, 2009.
[35] Nazari-Shirkoushi, S., Eivazy, H., Ghodsi, R., Rezaie, K., and Atashpaz-Gargari, E., "Solving the Integrated Product Mix-Outsourcing Problem using the Imperialist Competitive Algorithm", Expert Systems with Applications, Vol. 37, 2010, pp. 7615-7626.
[36] Hussain, M., Rajendran, Z., A., Kumar, Sh., and Kumar, G., "Comparison of PID Controller Tuning Methods with Genetic Algorithm for FOPTD System", Journal of engineering research and applications, Vol. 4, No. 2, 2014, pp. 308-314.