SRTTU

JCARME

ISSN: 2228-7922

Journal of Computational and Applied Research in Mechanical Engineering
Vol. 7, No. 1, pp. 23-34
jcarme.srttu.edu

Optimizing control motion of a human arm with PSO-PID controller

Mohammad Hossein Bayati Chaleshtari[1], Elaheh Norouzi [2] and Habib Ahmadi[*, 3]

[1] Young Researchers and Elite Club, Islamshahr Branch, Islamic Azad University, Islamshahr, Iran
[2] Amirkabir University of Technology, Biomedical Engineering Department, Tehran, Iran
[3]Mechanical Engineering Department,Shahrood University of Technology, Shahrood, Iran

Article info:

Received: 17/02/2016
Accepted: 13/12/2016
Online: 29/07/2017

Keywords:

Functional electrical stimulation (FES),
PID controller,
Particle swarm optimization algorithm,
Transverse plane.

Abstract

Functional electrical stimulation is the most commonly used system for restoring function after spinal cord injury. In this study, a model consists of a joint, two links with one degree of freedom, and two muscles as flexor and extensor of the joint, which simulated in MATLAB using Sim-Mechanics and Simulink Toolboxes, is used. The muscle model is based on Zajac musculotendon actuator and composed of a nonlinear recruitment curve, a nonlinear activation-frequency relationship, calcium dynamics, fatigue/recovery model, and an additional constant time delay, force -length and force-velocity factors. A classic controller for regulating the elbow joint angle; a Proportional- Integral- Derivative (PID) controller, is used. First, the PID coefficients are tuned using trial and error method, and then a particle swarm optimization (PSO) algorithm was used to optimize them. The important features of this algorithm include flexibility, simplicity, short solution time, and the ability to avoid local optimums. This PSO-PID controller uses the PSO algorithm to get the required pulse width for stimulating the biceps to reach the elbow joint to the desired angle. The fitness function is defined as sum square of error. The results of PSO -PID controller show the faster response for reaching the range of the set point than the PID controller tuned by trial and error. However, the PSO -PID is much better in terms of the rise time and the settling time, although the PID tuned by trial and error has no overshoot. The time to reach the zero steady state error is half in PSO -PID in comparison to PID tuned by trial and error.

1. Introduction

Individuals with C5/C6 level spinal cord injury (SCI) lose voluntary control of almost all muscles of the upper extremity. A neuroprosthetic system like Functional Electrical Stimulation (FES) can be used to restore paralyzed motor functions caused by SCI [1, 2]. FES artificially induces short electrical pulses to generate muscle contraction. FES can also be used to induce joint movement by stimulating the flexor and/or extensor muscles of the joint. Each joint is actuated by at least two muscle groups, flexor muscles, and

*Corresponding author
email address: habibahmadif@shahroodut.ac.ir

extensor muscles. Moreover, the maximum force that can be exerted by a muscle is a function of its length and the rate of change of its length, both of which can vary with joint angle. The tension produced in electrically stimulated muscle depends on the intensity and frequency of stimulation. The stimulation intensity is a function of the total charge transferred to the muscle, which depends on the pulse amplitude, duration, and frequency as well as the shape of the pulse train. The resulting torque about the joint that is actuated by the muscle depends on the tension in the flexor and extensor muscles as well as factors such as the biomechanics of the joint. The angle of a joint, or, alternatively, the torque produced about a joint, can be regulated by varying the tension produced in the flexor and extensor muscles of the joint. Consequently, the joint angle or joint torque can be controlled by modulating the pulse amplitude, pulse duration or frequency of stimulation. Typically either the pulse duration or the amplitude of stimulation is controlled [3-5].

In this paper, it is proposed to use a proportional-integral-derivative (PID) controller for regulating elbow angle in a negative feedback loop to compensate the error between the desired angle and the real one; and then optimize the PID parameters using a particle swarm optimization (PSO) algorithm. The controller is based on a nonlinear musculoskeletal model of the elbow joint's response to electrical stimulation of the biceps. It should be noted that the value of coefficients PID controller is considered as the objective function of PSO algorithm.

2.The model

2. 1.The musculoskeletal model

The musculoskeletal model that is used in this study, is a two-dimensional model of the arm, in the Transverse plane. It includes two muscles and one degree of freedom (elbow flexion-extension). The range of the elbow angle is from 0 to 160° [6]. The body segment and joint parameters for the model were obtained from cadaver studies by Zatsiorsky [7]. These parameters include the position of joint centers, inertia and mass parameters for body segments. The muscles concerned here are biceps and triceps, a pair of antagonist's muscles functioning as flexor and extensor of the elbow joint. The biceps has two heads; long and short heads. The triceps muscle group has three heads: lateral, medial and long heads. As it is focused on single joint movement control in a 2D plane, only uni articular muscles of elbow joint would be considered. Therefore, biceps long head (LH) and triceps lateral head (LtH) are selected for FES control. The simplified musculoskeletal model is illustrated in Fig. 1 [8].

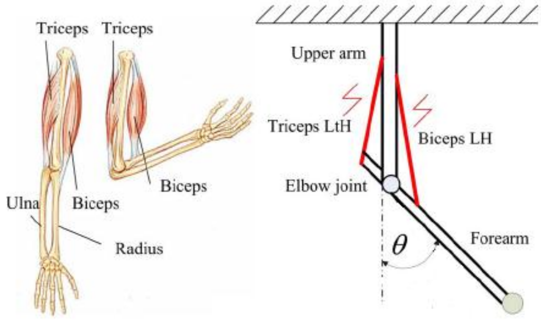

Fig. 1. Left graph is the physiological model of the elbow joint. Right graph is the simplified musculoskeletal model of the elbow joint [8].

2. 2.The muscle model

The muscle model is based on Zajac musculo-tendon actuator (Fig. 2) which accounts for the static and dynamic properties of both muscle and tendon. In this model, the muscle response to stimulation signal is composed of two parts: activation dynamics, and contraction dynamics [9].

2. 2. 1. Activation dynamics

When the muscle is stimulated by the electrical pulses, there is a dynamic process for the muscle to generate the force. This electrical characteristic of the muscle is named as activation dynamics [8-10]. Muscle activation is composed of the effect of spatial and temporal summation by a nonlinear recruitment curve, a nonlinear activation-frequency relationship, and calcium dynamics. A fatigue/recovery model and an additional constant time delay have been incorporated (Fig. 3).

2. 2. 2. Muscle recruitment curve

It can be modeled by a piecewise function with two values: a threshold pulse width (recruit deadband), and a saturation pulse width. The pulse width of the electrical pulse is defined as *z*, the normalized muscle recruitment curve a_r can be described in the following way:

$$a_r = \begin{cases} 0 & z \le PW_{thr} \\ \dfrac{1}{PW_{sat} - PW_{thr}}(z - PW_{thr}) & PW_{sat} \le z \le PW_{thr} \\ 1 & z \ge PW_{sat} \end{cases} \tag{1}$$

2. 2. 3 Frequency characteristic

When the frequency of stimulation pulse varies, it also affects the force produced by the muscle. This effect is defined by:

$$q(f) = \frac{(af)^2}{1+(af)^2} \tag{2}$$

where, q is the characteristic factor of the stimulation frequency.

2. 2. 4. Calcium dynamics

The muscle cannot be activated and relaxed simultaneously, and there exists a time delay. It can be modeled as a first order differential equation.

$$\dot{a} = \frac{1}{\tau_{ac}}(u^2 - ua) + \frac{1}{\tau_{ad}}(u - a) \tag{3}$$

where, a is the muscle activation without fatigue, $u = a_r q, \tau_{ac}$ is the activation time constant and τ_{da} is the de-activation time constant.

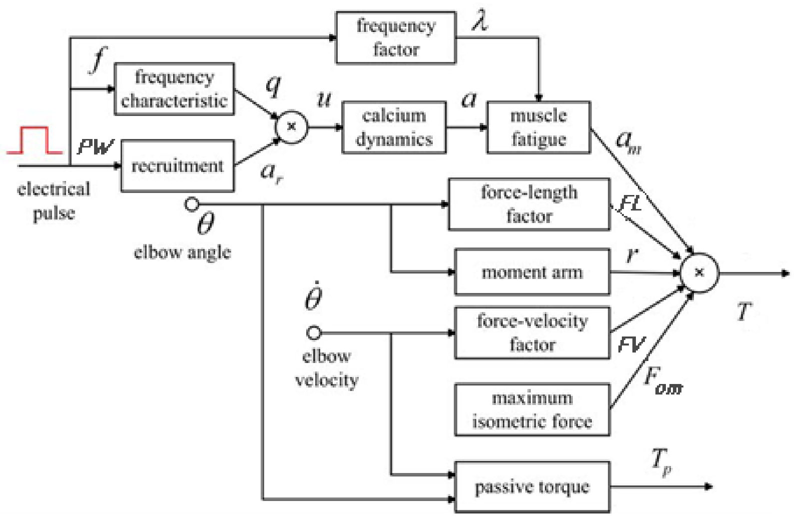

Fig. 2. Block diagram of muscle model based on Zajac muscle - tendon actuator [9].

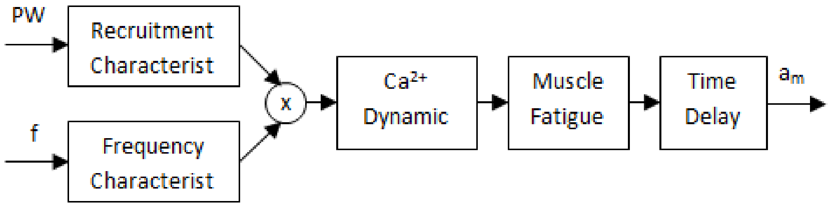

Fig. 3. Block diagram of the activation dynamics.

2.2.5. Muscle fatigue

When stimulating the muscle electrically, the force generated by the muscle drops as the time increases. This phenomenon is due to muscle fatigue. It depends on the activation level, *a*, and frequency, *f*, of the stimulation.

$$\frac{dp}{dt} = \frac{(p_{\min} - p)a\lambda(f)}{\tau_{fat}} + \frac{(1-p)(1-a\lambda(f))}{\tau_{rec}} \quad (4)$$

$$\lambda(f) = 1 - \beta + \beta(\frac{f}{100})^2 \quad (5)$$

where, P is the fatigue, τ_{fat} is the fatigue time constant, τ_{rec} is the recovery time constant, $P_{\min}$ is the minimum fitness, λ is the frequency factor on fatigue, and β is the shaping factor.

2.2.6. Contraction dynamics

Muscle contraction property is derived from the mechanical structure of the muscle (Fig. 4).

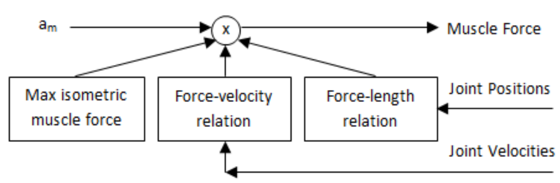

Fig. 4. Block diagram of the contraction dynamics

2.2.7. Force-length factor

A Gaussian-like function is used to model the relationship between the muscle force and length.

$$f_1 = \exp\left[-\left(\frac{(l-1)}{\varepsilon}\right)^2\right] \quad (6)$$

where, f_l is a normalized factor that describes the relationship between the muscle force and muscle length, l is the normalized muscle length with respect to the optimal muscle length: $l = l_m / l_{opt}$. The muscle active force strongly depends on the muscle length l_m . The peak force $F_{\max}$ occurs at the optimal muscle length l_{opt} .

2. 2. 8. Force-velocity factor

The muscle velocity also has an effect on the muscle force, and the factor f_v is used to describe this relationship.

$$f_v = 0.54\tan^{-1}(5.69v + 0.51) + 0.745 \quad (7)$$

where, v is the normalized muscle velocity with respect to the maximum contraction (shortening) velocity $v_{\max}$ of the muscle: $v = v_m / v_{\max}$. However, the muscle length lm and muscle velocity v_m are very difficult to be measured directly during real experiment. From a macroscopic point of view, it can be calculated by the joint angle and angular velocity. The relationship is defined as:

$$l_m = r(\theta - \theta_r) \quad (8)$$

$$v_m = r\dot{\theta} \quad (9)$$

where, θ and θ are the elbow joint angle and angular velocity, respectively. θ_r is the rest angle, and *r* is the muscle moment arm. Then, the force *F* produced by a muscle is defined as the product of maximum muscle force $F_{\max}$ with the dimensionless quantities f_l , f_v and a_m ,

$$F = F_{\max} \times f_l \times f_v \times a_m \quad (10)$$

where, a_m is the muscle activation with fatigue, $a_m = a_p$.

2. 2. 9. Passive torque

The passive torque is derived from the passive element in the muscle model. For the elbow joint, it is modeled as:

$$T_P = -0.2\dot{\theta} - 7.8\times10^{-7}\,\mathrm{sgn}(\theta - \frac{\pi}{2})\left[\exp\frac{36}{\pi\left|\theta - \frac{\pi}{2}\right|} - 1\right] \quad (11)$$

where, *sgn()* is the signum function specifying the sign of its argument [11, 12].

3. PSO algorithm

PSO is a heuristic global optimization method put forward originally by Doctor Kennedy and Eberhart in 1995 [13, 14]. It is developed from swarm intelligence and is based on the research of bird and fish flock movement behavior. While searching for food, the birds are either scattered or go together before they locate the place where they can find the food. While the birds are searching for food from one place to another, there is always a bird that can smell the food very well, that is, the bird is perceptible of the place where the food can be found, having the better food resource information. Because they are transmitting the information, especially the good information at any time while searching the food from one place to another, conduced by the good information, the birds will eventually flock to the place where food can be found. As far as PSO algorithm is concerned, solution swarm is compared to the bird swarm, the birds' moving from one place to another is equal to the development of the solution swarm, good information is equal to the most optimist solution, and the food resource is equal to the most optimist solution during the whole course [15, 16]. The most optimist solution can be worked out in PSO algorithm by the cooperation of each individual. The particle without quality and volume serves as each individual, and the simple behavioral pattern is regulated for each particle to show the complexity of the whole particle swarm. In the basic PSO algorithm, particle swarm consists of "n" particles, and the position of each particle stands for the potential solution in d-dimensional space. The particles change its condition according to the following three principles:

1) to keep its inertia
2) to change the condition according to its most optimist position
3) to change the condition according to the swarm's most optimist position.

The position of each particle in the swarm is affected both by the most optimist position during its movement (individual experience) and the position of the most optimist particle in its surrounding (near experience). When the whole particle swarm is surrounding the particle, the most optimist position of the surrounding is equal to the one of the whole most optimist particle; this algorithm is called the whole PSO. If the narrow surrounding is used in the algorithm, this algorithm is called the partial PSO. Each particle can be shown by its current speed and position, the most optimist position of each individual and the most optimist position of the surrounding. In the partial PSO, the speed and position of each particle change according to the following equality [17, 18]:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1^k (P_{Lbest\,id}^{\;k} - x_{id}^{k}) + c_2 r_2^k (P_{Gbest\,id}^{\;k} - x_{id}^{k}) \quad (12)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (13)$$

There are three main stages of a PSO algorithm, these are known as inertia weights, convergence factor, and selection.

3.1. Inertia weights

Inertia weights are put forward by some researchers [19, 20]. An Inertia weight ω is a proportional agent that is related to the speed of last time. The influence that the last speed has on the current speed can be controlled by inertia weights. The bigger the ω is, the bigger the PSO's searching ability for the whole is; and the smaller the ω is, the bigger the PSO's searching ability for the partial is. In this article, ω is equal to 1. Experimental results show that PSO has the biggest speed of convergence when ω is between 0.8 and 1.2. While experimenting, ω is confined from 0.9 to 0.4 according to the linear decrease, which makes PSO search for the bigger space at the beginning and locate the position quickly where there is the most optimist solution. As ω is decreasing, the speed of the particle is also slowing down to search for the delicate partial. The method quickens the speed of the convergence, and the function of the PSO is improved. When the problem that is to be solved is very complex, this method makes PSO's searching ability for the whole at the later period after several generations are not adequate, the most optimist solution cannot be found, so the inertia weights can be used to work out the problem [21].

3.2. Increase convergence factor

A PSO algorithm with convergence agents is introduced in references [22, 23], and the following is the formula for its position (χ) changing:

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \tag{14}$$

ϕ is called the convergence factor, $\phi = c_1 + c_2 >$ 4. Generally, ϕ is equal to 4.1, so χ is equal to 0.729.

The experimental result shows that the convergence speed in the PSO algorithm with the convergence agent compared with the PSO algorithm with inertia weights, is much quicker. In fact, when the proper ω , c_1 and c_2 is decided, the two calculation methods are identical. So, the PSO algorithm with convergence agent can be regarded as a special example of the PSO algorithm with inertia weights. Meanwhile, the properly selected parameters in the algorithms can improve the function of the methods.

3 .3. Selection

The compound PSO put forward by Angeline is based on the basic mechanism and the selection mechanism created during the development of the computers [24]. Due to POS's depending on P_{Lbest} and P_{Gbest} during its searching, the area to be searched is confined greatly. The introduction of the selection mechanism solves the problem gradually. The test result shows although selection has a better effect than basic PSO in the most tested functions, the result is less satisfying as far as the function "Griewank" is concerned. As a result, this method improve PSO's searching ability for the partial, meanwhile, it makes the searching for the whole area less powerful.

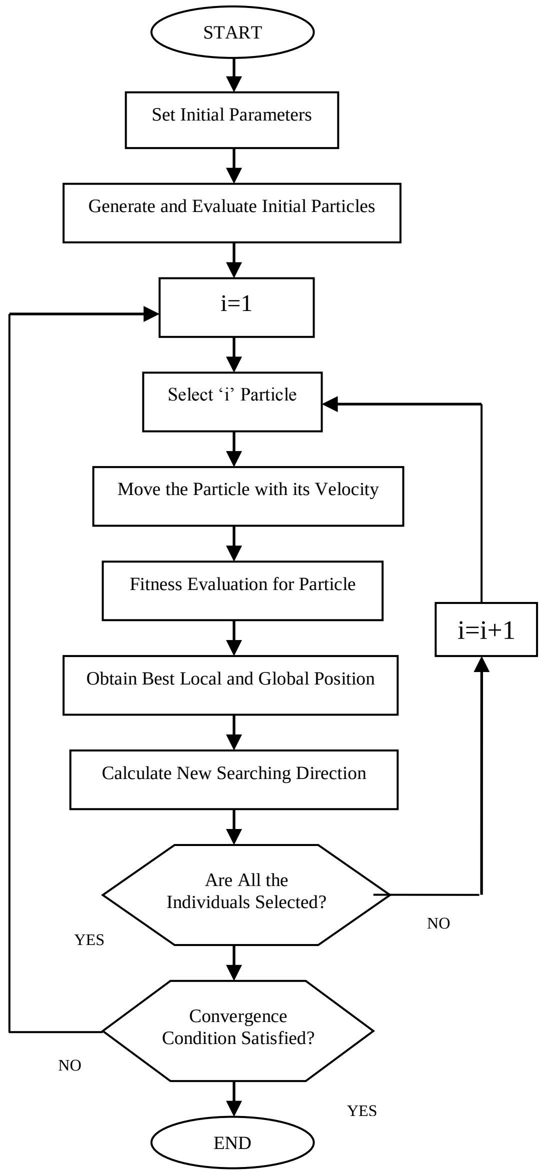

Fig. 5. PSO algorithm process flowchart [19]

4. Hand control in the transverse plane

This section examines the control flexion and extension movements of the elbow in the Transverse plane. Then coordinates of the model is defined in the Transverse plane. The authors want to the PID controllers to control flexion and extension movements of the hand to do it. In transverse plane, the path feed-forward control is not used because based on the available experiences, the authors conclude that the addition of this path to feedback, not only does not improve the system but it causes the system to be weaker. In Transverse plane, dealing with the gravity in the model is added to the controlling. The feedback path in control system is for reaching to desired range, feed-forward control can hold the arm in the desired angle. But to control the Transverse plane, only the PID controllers are used in the feedback path. Moreover, to have a better vision of correct movement in the Transverse plane, data model are applied to a model available in opensim software which provides the desired movement. One of the activities of daily living (ADL) is stimulated in MATLAB software by modeling 2 links, 2 joints and 7 muscles. The kinematic results were recorded of this model in MATLAB and then processed and applied to a model in opensim software. The results of this special task are shown in Fig. (6).

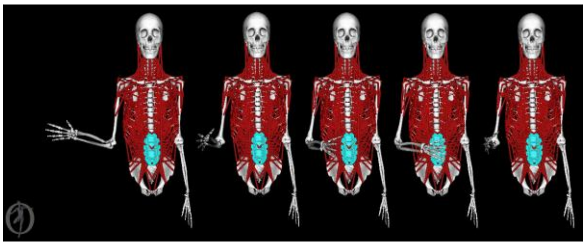

Fig. 6. Observation of arm movement in Transverse plane and application of the data record in the simulink model and using a kinematic model for opensim software.

4.1. The PID controller in the transverse plane

In this study, a classic controller is used to regulate the elbow angle; a proportional-integral- derivative controller. The proportional controller output uses a 'proportion' of the system error to control the system. However, this introduces an offset error into the system. The integral controller output is proportional to the amount of time. The integral action removes the offset introduced by the proportional control but it introduces a phase lag into the system. The derivative controller output is proportional to the rate of change of the error. Derivative control is used to reduce/eliminate overshoot and introduce a phase lead action that removes the phase lag introduced by the integral action. Three types of control are combined together to form a continuous PID controller (Fig. 7) with the transfer function as follows:

$$C_{PID}(s) = \frac{K_D s^2 + K_P s + K_I}{s} \tag{15}$$

So the PID controller generates muscle stimulations that are proportional to the errors in joint angles and their time derivatives and time-integrals. The controller is placed in a negative feedback loop to compensate the error between the desired elbow angle and the real one.

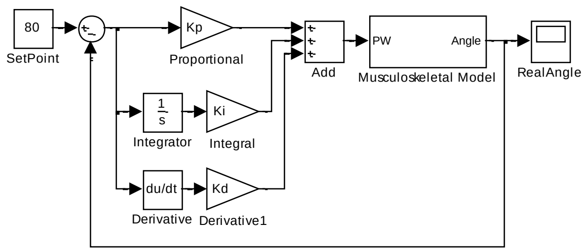

Fig. 7. Block diagram of Proportional- Derivative-Integral control of the musculoskeletal model with set point of 80 degrees.

4. 2. Optimizing PID controller using PSO algorithm

The PSO method is based on swarm intelligence. The research on it is just at the beginning. Apart from the PSO and the simulated annealing (SA) approach, the POS has no systematical calculation method and it has no definite mathematic foundation. The research on PSO concerns mainly the mathematic foundation and application research [25, 26]. The mathematic foundation includes the mechanical principle of PSO itself, the proof of its convergence, and robustness and etc. So it expresses the advantages of PSO algorithm as follows:

1) PSO is based on the intelligence. It can be applied to both scientific research and engineering use.

2) PSO have no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast.

3) The calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability and it can be completed easily.

4) PSO adopts the real number code, and it is decided directly by the solution. The number of the dimension is equal to the constant of the solution.

The PID controller parameters are firstly identified by trial and error and then optimized using a PSO algorithm. The initial guess of the optimization is derived by adding a uniform random number in a range of [0, 1] to the coefficients tuned by trial and error. Then a further adjustment is performed with a PSO algorithm. The PSO-PID controller uses PSO algorithm to get the required pulse width for stimulating the biceps to reach the elbow joint to the desired angle [27].

As shown in Fig. 2, new theta (elbow angle) can be calculated using each block formula. It is already known the relation between the error (as the input of PID controller) and controller's output as the pulse width for stimulating the muscle. In this section, the parameters used in PSO algorithm are explained. One of the important steps in PSO is the determination of population number. There is no fast and thumb rule with regards to which is the best method to adopt. For a long time, the decision on the population size is based on trial and error. In this project, an initial population of 80 is used since there are three variable parameters. The convergence criterion of a PSO algorithm is a user-specified condition. PSO uses four different criteria to determine when to stop the solver. Another criterion is the maximum limit of velocity. v_{id}^{k} and x_{id}^{k} stand separately for the speed of the particle "i" at its "k" times and the d-dimension quantity of its position; $P_{Lbest\,id}^{\;k}$ represents the d-dimension quantity of the individual "i" at its most optimist position at its "k" times, respectively. $P_{Gbest\,d}^{\;k}$ is the d-dimension quantity of the swarm at its most optimist position. In order to avoid particle being far away from the searching space, the speed of the particle created at its each direction is confined between $-v_{d\max}$, and $v_{d\max}$. If the number of $v_{d\max}$ is too big, the solution is far from the best, if the number of $v_{d\max}$ is too small, the solution is the local optimism. Finally, max limit of velocity value is used equal to 0.1. The fitness (objective) function is used to provide a measure of how individuals are performed in the problem domain. In the case of minimizing problem, the fittest individuals have the lowest numerical value of the associated objective function. This raw measure of fitness is only used as an intermediate stage in determining the relative performance of individuals in a PSO. c_1 and c_2 is other criterion that represents the speeding figure, regulating the length when flying to the most particle of the whole swarm and to the most optimist individual particle. If the figure is too small, the particle is probably far away from the target field. But if the figure is too big, the particle may fly to the target field suddenly or fly beyond the target field. The proper figures for c_1 and c_2 control the speed of the particle's flying and the solution is not the partial optimism. In this article, c_1 equals to c_2 and they equal to 2 and r_1 and r_2 are random numbers between (0-1). Moreover, weight parameter is equal to 1. Another main operator is the selection. Griewank selection is chosen by the authors. Griewank selection has a better effect than basic PSO in the most tested functions. As a result, this method improves PSO's searching ability for the partial. It creates the searching for the entire domain less powerful [28].

5. Results and discussion

In this section, the PID feedback control is primarily used to control the movement of flexion and extension in the plane Transverse control use. For flexion movement, initial angle and angle goal on 10 to 80 degrees, respectively and such as control in the Transverse plane, the controller outputs are separated into positive and negative. The positive part of the pulse width applied to the biceps muscle and the negative part, with positive sign applied to triceps muscle. The controller coefficients are set by trial and error, and then a PSO algorithm is used to optimize the coefficient.

$$K_P = 3, \qquad K_i = 4, \qquad K_d = 0.2$$

Then, the result of PID control for flexion movement (from 10 to 80 degrees) in the Transverse plane is in two cases of:
A) set by trial and error
B) optimized by PSO algorithm.
which are provided in Fig. 8. The angle of the system output is represented in Fig. 8(a). It shows changing the angle of the elbow angle from 10 degrees to the desired angle of 80 degrees. Figure 8(b) provides the pulse width controller output (during 20 seconds).
The results obtained from controller response are compared:
The rise time and settling time were measured in the state (a) were 1.01 and 2.05, respectively, while in (b) they were 0.24 and 0.41. The overshoot and steady state error in (a) were equal to zero but in (b) the amount of overshoot was equal to 0.08.
Considering a very small overshoot and steady state error in the controller which is optimized by PSO algorithm, the conveying speed of the elbow angle from 10 to 80 degrees would be much faster.

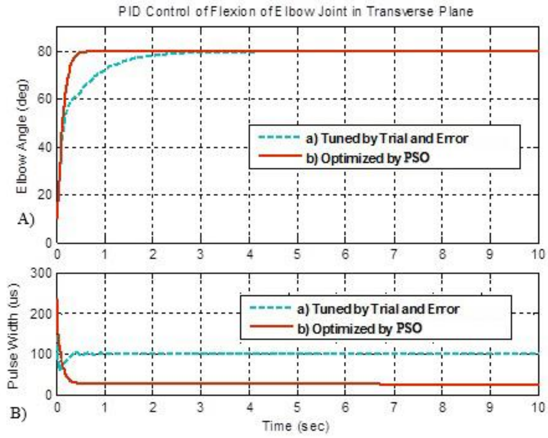

Fig. 8. PID controller to move the elbow flexion in the Transverse plane with coefficients of (a) trial and error, and (b) optimized with PSO algorithm. In A) elbow angles were compared. In B) pulse width were compared.

6. Conclusions

The aim of the present paper is to improve the control performance using a PSO algorithm for PID controller tuning. The most important feature of this algorithm includes high performance in searching the solution space, flexibility and the ability to avoid local optimums. Unlike single-trajectory methods, this method searches the solution space globally; therefore, it is less likely to get trapped in local optimums. Testing of the quality control process is accomplished in simulation environment of MATLAB software using Simulink and Sim-mechanics toolboxes on a nonlinear musculoskeletal model. In conclusion, the responses show that the PSO-PID controller has much faster response than the classical one. The classical PID values are good for providing the starting population of the PSO. Also, there are many steps beside trial and error in getting the PID values before it can be narrowed down in getting close to the "optimized" values. An optimized algorithm is implemented in the system to see and study how the system response. However, the PSO-PID controller is much better in terms of the rise time and the settling time than the classical PID, which has no overshoot, although it suffers in terms of rising time and settling time. With respect to the computational time, it is noticed that the classical PID takes a longer time to reach its peak as compared to the PSO-PID controller. The controller with coefficients is tuned for a desired angle and is tested for a wide range of set points. The results are great for a wider range in PSO-PID than the classical PID. Generally, meta-heuristic algorithms based on particle collections are a type of random search on the basis of population. The first and most important advantage of these algorithms is that they are inherently parallel and they can examine search space in different directions. Parallel examination under spaces causes space searching head to the areas in which statistical average of the objective function is high, and the presence of absolute optimum point is more possible. This is because, unlike one-way methods, the solution space is searched thoroughly and it is unlikely possible to be caught in a local optimum point.

7. Acknowledgment

The authors are indebted to Ali Fallah, the associate professor in Department of Biomedical Engineering, Amirkabir University of Technology for his valuable and constructive suggestions and encouraging comments, which significantly improved the quality of the manuscript.

References

[1] C. C. L. Lynch, and M. R. Popovic, "Closed-Loop Control for FES: Past Work and Future Directions", *10th Annual Conference of the International FES Society*, Ottawa, Canada, pp. 56-59, (2005).

[2] S. Bin Mohamed Ibrahim, "The PID Controller Design Using Particle swarm optimization Algorithm", PhD thesis, University of Southern Queensland, Toowoomba, Australia, pp.132-137, (2005).

[3] D. Blana, E.K. Chadwick, A. Bogert and R. F. Kirsch, "Feedback Control for a High Level Upper Extremity Neuroprosthesis", *ASB 29th Annual Meeting*, Cleveland, Ohio, pp. 28-33, (2005).

[4] C. L. Lynch and M. R. Popovic, "Functional Electrical Stimulation", *IEEE Control Systems MaGAzine*, Cleveland, Ohio, (2008).

[5] D. Blana, R. F. Kirsch and E. K. Chadwick, "Combined Feed forward and Feedback Control of a Redundant, Nonlinear, Dynamic Musculoskeletal System", *International Federation for Medical and Biological Engineering*, Vol. 47, No. 5, pp. 533-542, (2009).

[6] M. Ferrarin, F. Palazzo, R. Riener and J. Quintern, "Model-Based Control of FES-Induced Single Joint Movements", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 9, No. 3. pp. 78-88, (2001).

[7] V. M. Zatsiorsky, "Kinetics of Human Motion", Human Kinetics, New Zealand, pp. 265-350, (2002).

[8] D. Zhang, and W. T. Ang, "Tremor Suppression of Elbow Joint via Functional Electrical Stimulation: A Simulation Study", *Proceeding of the 2006 IEEE, International Conference on Automation Science and Engineering*, Beijing, China, pp.76-81 (2006).

[9] A. Maleki and R. Shafaei, "Musculoskeletal Model of Arm for FES Research Studies", *4th Cario International Biomedical Conference*, University of Salford, Ireland UK, (2008).

[10] Negin Hesam-Shariati, "Control of Reanimation of Paralyzed Arm for Reaching Movement Using FES'', M. Sc. Thesis, Bioelectrics, Amirkabir University of Technology, 183 pages, (2012).

[11] K. Kurosawa, R. Futami, T. Watanabe and N. Hoshimiya, "Joint Angle Control by FES Using a Feedback Error Learning Controller", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 13, No. 3, pp. 62-77, (2005).

[12] M. O. Ali, S. P. Koh, K. H. Chong, S. K. Tiong and Z.A. Obaid, "Genetic Algorithm Tuning Based PID Controller

for Liquid-Level Tank System", *Proceedings of the International Conference on Man- Machine Systems (ICoMMS)*, Kuala Lumpur, Malaysia, pp. 29-33 (2009).

[13] Eberhart R, and J. Kennedy, A New Optimizer Using Particle Swarm Theory. Proc of *6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan. IEEE Service Center Piscataway NJ, pp. 39-43, (1995).

[14] J. Kennedy, R. Eberhart, Particle Swarm Optimization. Proc of IEEE International Conference on Neural Network, Perth, Australia, *IEEE Service Center Piscataway NJ*, pp. 1942-1948, (1995).

[15] M. Clerc, and J. Kennedy, The particle swarm-explosion stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp. 58-73, (2002).

[16] M. Dorigo, V. Maniezzo, and A. Colorni, Ant system: optimization by a colony of cooperationg agents. *IEEE Transactions on Systems. Man*, and Cybernetics-Part B, Vol. 26, No. 1, pp. 29-41, (1996).

[17] Y. Shi, and R. C. Eberhart, A modified particle swam optimizer. *IEEE Word Congress on Computational Intelligence*, pp. 69-73, (1998).

[18] Zheng Jianchao, Jie Jing, and Cui Zhihua., Particle swam optimization. *Science Publishing Company of Beijing*. Vol. 22, No. 3, pp. 94-112 (2004).

[19] R. C. Eberhart, and Y. Shi, Comparison between genetic algorithms and Particle Swarm Optimization. Porto V W, Saravanan N, Waagen D, et al. Evolutionary Programming VII. [S.l.]: Springer, pp. 611-616, (1998).

[20] R. C. Eberhart, and Y. Shi, Comparing inertia weights and constriction factors in Particle Swarm Optimization. *Proceedings of the Congress on Evolutionary Computation*, pp. 84-88, (2000).

[21] N. Higashi, and H. Iba, Particle Swarm Optimization with Gaussian mutation. *Proceedings of the 2003 Congress on Evolutionary Computation. Piscataway, NJ*: IEEE Press, pp. 72-79, (2003).

[22] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Service Center, pp. 1951-1957, (1999).

[23] A. Colorni, M. Dorigo, and V. Maniezzo, et al.; Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life, pp. 134-142, (1991).

[24] P. J. Angeline, Using selection to improve Particle Swarm Optimization. Proceedings of the *Congress on Evolutionary Computation. Piscataway*. NJ: IEEE Press, pp. 84-89, (1999).

[25] Duan Haibin, Ant Colony Optimization theory and Application. *Science Publishing Company of Beijing*. Vol. 23, No. 1, pp. 82-93 (2005).

[26] Gao Ying, Xie Shengli, Particle swam optimization Algorithm based on simulated annealing (SA) approach .*Computer Engineering and Application*, Vol. 40, No. 1, pp. 47-49, (2004).

[27] Qinghai Bai, Analysis of Particle Swarm Optimization Algorithm *'Journal of Computer and Information Science'*, Vol. 3, No. 1, pp. 57-71, (2010).

[28] Chen Yonggang, Yang Fengjie, and Sun Jigui.; A new Particle swam optimization Algorithm. *Journal of Jilin University*, Vol. 24, No. 2, pp. 181-185, (2006).

How to cite this paper:

Mohammad Hossein Bayati Chaleshtari, Elaheh Norouzi, and Habib Ahmadi, *"Optimizing control motion of a human arm with PSO-PID controller*", *Journal of Computational and Applied Research in Mechanical Engineering,* Vol. 7. No. 1, pp. 23-34

DOI: 10.22061/JCARME.2017.641

URL: http://jcarme.srttu.edu/article_645.html