[1]Biomedical Engineering Faculty, Amirkabir University of Technology, Tehran, Iran.
[2]Electrical and Computer Engineering Faculty, Semnan University, Semnan, Iran.

Genetic Feedforward-Feedback Controller for Functional Electrical Stimulation Control of Elbow Joint Angle

Hesam Shariati N.[1]*, Maleki A.[2], Fallah A.[1]

Abstract

Background: Functional electrical stimulation (FES) is the most commonly used system for restoring functions after spinal cord injury (SCI).

Objective: In this study we investigated feedback PID and feedforward-feedback P-PID controllers for regulating the elbow joint angle.

Methods: The controllers were tuned based on a nonlinear muculoskeletal model containing two links, one joint with one degree of freedom and two muscles in the sagittal plane that was simulated in MATLAB using Sim Mechanics and Simulink toolboxes. The first tune of the PID and P-PID controllers was done by trial and error. Then, the coefficients were optimized by genetic algorithm (GA). For checking the robustness of the controllers, we compared the amount of rise time, settling time, maximum overshoot and steady state error under three conditions: the first was when the initial angle of the joint was fixed and only the desired angles changed; the second was with a fixed step as input and various initial angles; and the last condition was with different maximum forces for muscle.

Results: Genetic controllers had better performance than the trial and error tuned controllers. The amounts of settling time were not so different for the controllers in condition 1 but had more variations in condition 2 and had really better results in genetic P-PID in condition 3. The overshoot was pretty less in PIDs than in P-PIDs and the steady state error was almost zero for all of the controllers.

Conclusion: Genetic controllers had a better performance than the trial and error tuned controllers. The rise time was much less in P-PIDs than in PIDs.

Keywords

Functional electrical stimulation; Feedforward-feedback; PID controller; Genetic algorithm

Introduction

Individuals with spinal cord injury (SCI) at the level of C5-C6 lose voluntary control of almost all muscles of their upper extremity. In these patients, a neuroprosthetic system like functional electrical stimulation (FES) by stimulating paralyzed muscles to contract in appropriate patterns [1, 2], can be used to restore the impaired motor function.

FES generates short electrical pulses to create muscle contraction. FES can also be used to create joint movement by stimulating the flexor and extensor muscles of the joint. Each joint is actuated by at least two muscle groups—flexors and extensors. The maximum force that can be exerted by a muscle is a function of its length and the rate of change in its length, both of which can vary with joint angle. The tension produced in an electrically stimulated muscle depends on the intensity and frequency

*Corresponding author: Negin Hesam-Shariati, Biological Systems Control Laboratory, Biomedical Engineering Faculty, No. 424, Hafez Ave, Tehran, Iran.
Fax: +98-21-6646-8186
Tel: +98-21-6454-2350-31
E-mail: negin_hsh@aut.ac.ir

of the stimulation. The stimulation intensity is in turn 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 the stimulation. Typically, either the pulse duration or the amplitude of stimulation is controlled [4-6].

Lynch, *et al* [5], based on a nonlinear physiological model of the knee, tested four controllers for regulating knee angle. The control methods used were open-loop controller, closed-loop PID controller, feedforward-feedback controller and adaptive controller.

Feedback control monitors the output to make corrections and to make output to behave as desired [7]. Feedback has been used for a variety of FES applications. In addition to feedback controllers, a variety of techniques including combined feedforward and feedback control [2, 3] have been used. These highly tuned controllers are often comparable to linear PD and PID controllers [8] that may be suboptimal.

In this paper, two types of FES controllers are proposed: a proportional-integral-derivative (PID) controller and a combined feedforward-feedback controller for regulating elbow joint angle.

Materials and Methods

In our setup, while the feedforward controller generated the nominal muscle activations required for the desired movement, and the feedback controller corrected the errors caused by muscle fatigue and external disturbances. The feedforward controller used was a proportional (P) controller and the feedback controller was a PID controller.

The controllers were first tuned by trial and error, and then optimized by a genetic algorithm. The controllers' tuning was done using a musculoskeletal model of the elbow joint and two mono-articular muscles as flexor and extensor with one degree of freedom which was simulated in MATLAB using Simulink and SimMechanics toolboxes. The model was based on the elbow joint's response to electrical stimulation of the biceps.

The Model

The Musculoskeletal Model

The musculoskeletal model that was used in this study, was a two-dimensional model of the arm in the sagittal plane. It included two muscles and one degree of freedom—elbow flexion-extension. The range of the elbow angle was from 0° to 160° [3]. The body segment and joint parameters for the model were obtained from cadaver studies by Zatsiorsky [9]. These parameters included the position of joint centers, inertia and mass parameters for body segments.

The muscles concerned here were biceps and triceps—a pair of antagonist muscles functioning as the flexor and extensor of the elbow joint, respectively. The biceps has two heads: a long and a short head. Triceps has three heads: lateral, medial and long heads. As our focus was on single joint movement control in 2D plane, only uniarticular muscles of elbow joint were considered. Therefore, biceps long head (LH) and triceps lateral head (LtH) were selected for FES control. The simplified musculoskeletal model is illustrated in Figure 1 [10].

The Muscle Model

The muscle model was based on Zajacmusculo-tendon actuator which considers both 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

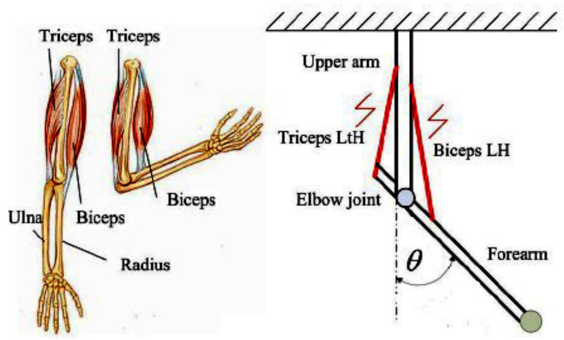

Figure 1: Left graph is the physiological model of the elbow joint. Right graph is the simplified musculoskeletal model of the elbow joint [10].

[11].

When a muscle is stimulated by electrical pulses, a dynamic process happens in the muscle that generates the force. This electrical characteristic of the muscle is named "activation dynamics" [10-12].

Muscle activation is composed of spatial and temporal summation that acts according to a nonlinear recruitment curve, a nonlinear activation-frequency relationship, and calcium dynamics. A fatigue/recovery model and an additional constant time delay were also incorporated (Fig. 2).

Muscle recruitment curve can be modeled by a piece-wise function with two values: a threshold pulse width (recruit deadband), and a saturation pulse width. If the pulse width of the electrical pulse is represented as z, the normalized muscle recruitment curve (a_r) can be described as follows:

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

Frequency characteristics: When the frequency of stimulation pulse varies, it also affects the force produced by the muscle. This can be described by the following equation:

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

where q represents the characteristic factor of the stimulation frequency.

Calcium dynamics: There is always a delay between muscle contraction and relaxation—these cannot be simultaneous. This phenomenon can be modeled as a first order differential equation:

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

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

Muscle fatigue: When stimulating a muscle electrically, the force generated by the muscle will drop by time. This phenomenon is due to muscle fatigue that depends on the activation level (a) and frequency (f) of the stimulation according to the following equa-

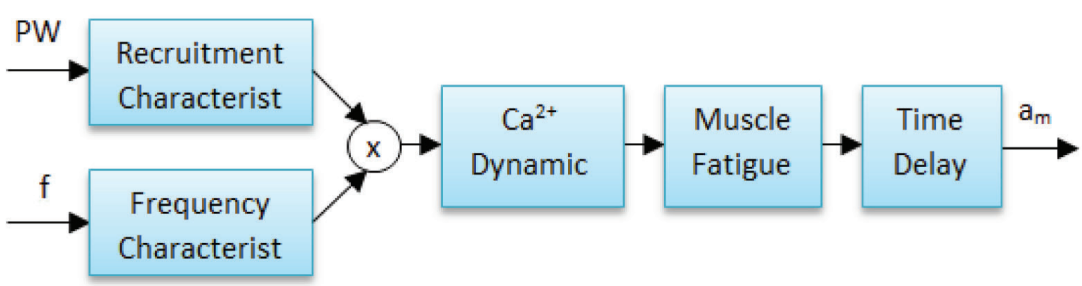

Figure 2: Block diagram of the activation dynamics

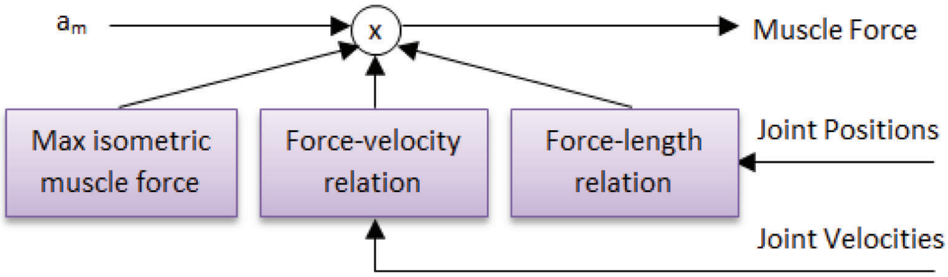

Figure 3: Block diagram of the contraction dynamics

tions:

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

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

where p represents the fatigue, τ_{fat} the fatigue time constant, τ_{rec} the recovery time constant, p_{min} the minimum fitness, λ the frequency factor on fatigue, and β is the shaping factor.

Contraction dynamics: Muscle contraction property originates from the mechanical structure of the muscle (Fig. 3).

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

$$f_l = e^{-\left(\frac{(l-1)}{\varepsilon}\right)^2}$$

where f_l is a normalized factor that describes the relationship between the muscle force and muscle length, and 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 l_m. The peak force, F_{max} occurs at the optimal muscle length, l_{opt}.

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 as follows:

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

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, direct measurement of muscle length, l_m and muscle velocity, v_m in real experiment are very difficult, but they can be calculated by measuring the joint angle and the angular velocity according to the following equations:

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

where θ_r is the rest angle, and r is the muscle moment arm; and

$$v_m = r\dot{\theta}$$

where θ and $\dot{\theta}$ are the elbow joint angle and angular velocity respectively.

The force, F produced by a muscle is then could be calculated as the product of maximum muscle force, F_{max} and the dimensionless factors f_l, f_v and a_m, as follows:

$$F = F_{\max} \times f_l \times f_v \times a_m$$

where a_m is the muscle activation with fatigue and equals to $a \times p$.

Passive torque originates from passive elements in the muscle and for the elbow joint (Fig. 4); it can be modeled as:

$$T_P = -0.2\dot{\theta} - 7.8\times10^{-7}\, sgn\left(\theta - \frac{\pi}{2}\right)\left(e^{\frac{36}{\pi\left|\theta - \frac{\pi}{2}\right|}} - 1\right)$$

where *sgn* is the *signum* function.

Genetic Algorithm

Genetic algorithm (GA) is a stochastic global search method that mimics the process of natural evolution. The GA starts with no knowledge of the correct solution and depends entirely on responses from its environment and

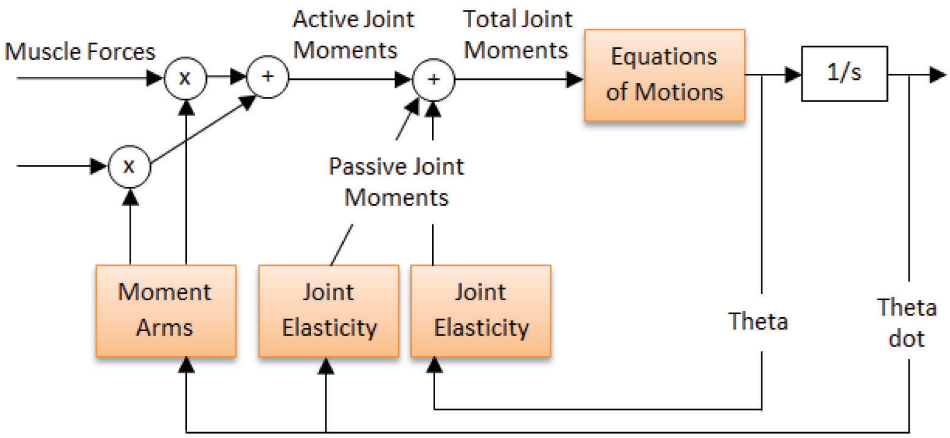

Figure 4: Block diagram of the activation dynamics

evolution operators to arrive at the best solution. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging to suboptimal solutions [13, 14].

A GA is typically initialized with a random population. This population is usually represented by a real-valued number or a binary string called a "chromosome." How well an individual performs a task is measured by an objective function. The objective function assigns a number, the so-called fitness to each individual. The fitness of each chromosome is then assessed and a survival of the fittest strategy is applied.

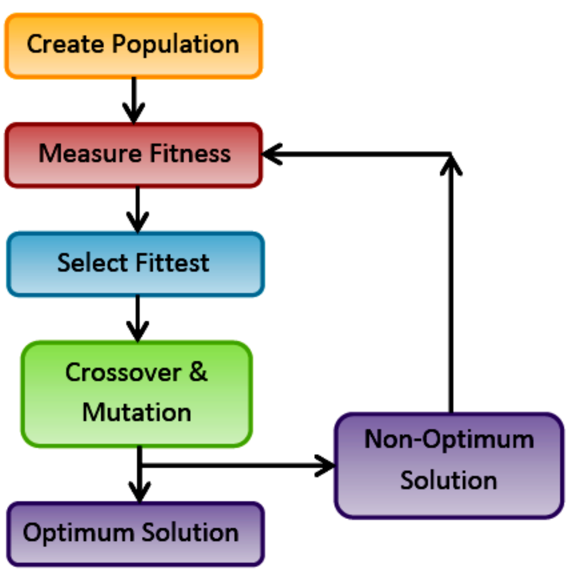

Figure 5: Genetic algorithm process flowchart

There are three main stages of a genetic algorithm. These are termed "reproduction," "crossover," and "mutation."

`Reproduction:` During the reproduction phase the fitness value of each chromosome is assessed. Just like in natural evolution, the probability of an individual being selected for reproduction is related to its fitness, ensuring that fitter individuals are more likely to leave offspring.

There are four common methods for selection:

1. Roulette wheel selection
2. Stochastic universal sampling
3. Normalized geometric selection, and
4. Tournament selection

All selection methods are based on the same principle that is giving fitter chromosomes a larger probability of selection. If this procedure is repeated until there are enough selected individuals, this selection method is called *roulette wheel* selection. If instead of a single pointer spun multiple times, there are multiple, equally spaced pointers on a wheel that is spun once, it is called *stochastic universal* sampling. Repeatedly selecting the best individual of a randomly chosen subset is *tournament selection.* Taking the best half, third or another proportion of the individuals is *geometric selection.*

`Crossover:` Once the selection process

is completed, the crossover algorithm is initiated. The crossover operation swaps certain parts of two selected strings in a bid to capture the good parts of old chromosomes and create better new ones. Genetic operators manipulate the characteristics of a chromosome directly, using the assumption that certain individual's gene codes, on average, produce fitter individuals. The crossover probability indicates how often crossover is performed. A probability of 0% means that "offspring" will be replicas just the same as their "parents" and a probability of 100% means that each generation will be composed of entirely new offspring.

Mutation: Using selection and crossover on their own will generate a large amount of different strings. However, there are two main problems with this:

Depending on the initial population chosen, there may not be enough diversity in the initial strings to ensure the GA searches the entire problem space.

The GA may converge on sub-optimum strings due to a bad choice of initial population.

These problems may be overcome by the introduction of a mutation operator into the GA. Mutation is the occasional random alteration of a value of a string position. The probability of mutation is normally low because a high mutation rate would destroy fit strings and degenerate the genetic algorithm into a random search.

Mutation probability values of around 0.01% to 0.1% are common; these values represent the probability that a certain string will be selected for mutation.

The process of GA used is summarized in Figure 5.

A general scheme of a GA can be described by the following steps [13]:

1. Initialization of the population of chromosomes (set of randomly generated chromosomes)
2. Evaluation of the fitness for all chromosomes
3. Selection of parent chromosomes (the fittest members of the population)
4. Crossover and mutation of the parents children
5. Completion of the new population from the new children and selected members of the old population (until a predefined convergence criterion is met)
6. Jump to the step 2.

Controller Optimization

The controller: There is different control strategies used in existing FES systems. Feedforward control has the advantage that no sensors are required to facilitate rapid movements and greatly simplifies controller implementation in humans. This kind of controller is generally used in clinical tasks and the output depends only on the user command, and not on the system performance. Another advantage of feedforward control is that corrective action is taken for a change in a disturbance input before it affects the control parameter. However, drawbacks include the inability to make corrections if the actual movement deviates from the desired one due to muscle fatigue or any changes in the environment, and the requirement to have detailed system behavior to produce an accurate movement [3, 15].

Feedback control uses sensors to monitor the system output, so it can correct when the output does not behave as desired. Feedback is necessary in order to maintain good tracking performance in the presence of fatigue and any external disturbances encountered. Challenges to the success of feedback control include limitations in sensor signal quality, the relatively slow response properties of muscles, and inherent delays in system response, which are of particular concern for fast movements.

In this study, at first we used a classic controller for regulating the elbow angle—a proportional-integral-derivative controller. The controller was placed in a negative feedback loop to compensate the error between the desired elbow angle and the actual one. So the PID

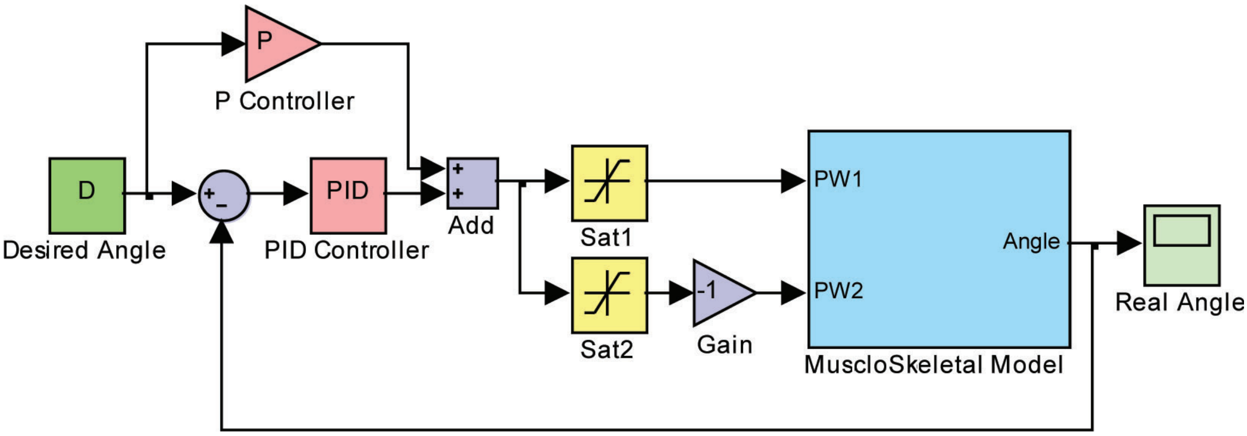

Figure 6: Block diagram of feedforward-feedback P-PID control of the musculoskeletal model

controller generates muscle stimulations that are proportional to the errors in joint angles and their time-derivatives and time-integrals.

In the next step, we added a feedforward path to the previous controller. Feedforward control is commonly used in clinical practice. The advantage of this design is that it does not need sensors to measure the system output, but the disadvantage is that it is unable to make corrections if the actual angle deviates from the desired angle.

Therefore, a combination of feedforward and feedback control has the best results, and is the preferred control method in several FES system design, including the one presented in this paper.

The feedforward controller is typically an inverse-dynamic model of the controlled system. Since an actual inverse-dynamic model is usually not available, here we used a proportional control as the feedforward controller.

The input of the controllers was the desired angle, and the output was pulse width needed for stimulating the biceps.

The FES controllers presented in this paper were designed and tested in simulation.

The controller optimization using GA: In optimizing, the controllers use GA to get the required pulse width for stimulating the biceps to bring the elbow joint to the desired angle.

The initial guess of the parameters was derived by adding a uniform random number ranging from zero to one to the coefficients tuned by trial and error; further adjustment was then performed with a GA.

One of the important steps in GA is determining the number of population. However, there is no fast and rule of thumb for determining which method is the best to adopt. The decision on the population size is usually based on trial and error. In this study, an initial population size of 20 was used since there were 3 or 4 variable parameters.

The convergence criterion of a genetic algorithm is a user-specified condition. GA uses four different criteria to determine when to stop the solver: GA stops when the maximum number of generations is reached. GA also detects when the string fitness value exceeds a certain threshold for some time given in seconds, or for some number of generations. Another criterion is the maximum time limit in seconds. In our study, GA stopped when the maximum number of generations exceeded 200 or when there was no change in the best fitness value for 150 generations.

The fitness (objective) function is used to provide a measure of how individuals have performed in the problem domain. In the case of a minimization problem, the fittest individuals will have the lowest numerical value of the associated objective function. This raw measure of fitness is usually only used as an intermediate stage in determining the relative performance of individuals in a GA. In this

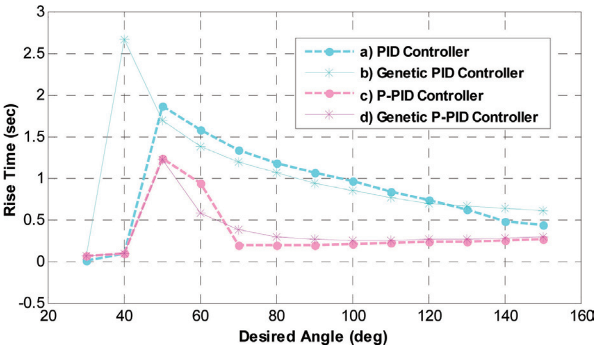

Figure 7: Comparison of the rise time for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same initial angle of 10° and various desired angles of 30°–150°

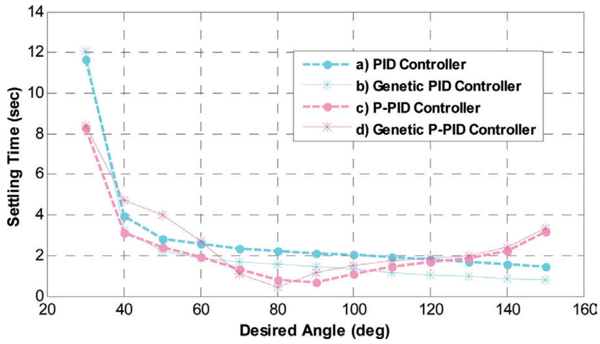

Figure 8: Comparison of the settling time for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same initial angle of 10° and various desired angles of 30°–150°

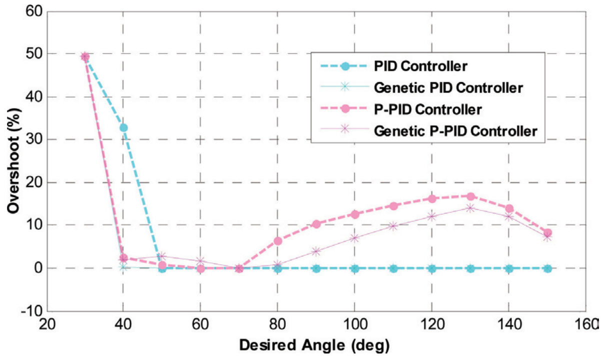

Figure 9: Comparison of the overshoot for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same initial angle of 10° and various desired angles of 30°–150°

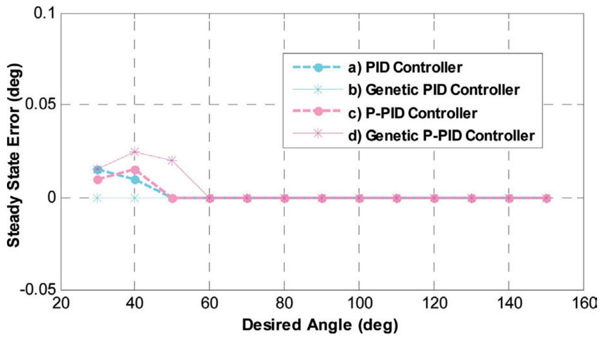

Figure 10: Comparison of the steady state error for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same initial angle of 10° and various desired angles of 30°–150°

study, sum square of errors was used to assess the fitness of each chromosome.

GA uses operators to produce the next generation of the population. The different operators are selection, crossover, and mutation.

Here we chose tournament selection. Tournament selection involves running several tournaments among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected. Tournament selection has several benefits: it is efficient to code, works on parallel architectures and allows the selection pressure to be easily adjusted.

Results

The four controllers described were tuned and optimized when the initial elbow angle was 10° and the desired angle was 80°. For checking the robustness of the controllers, we compared the amount of rise time, settling time, maximum overshoot and steady state error in three conditions for the controllers.

The first case was when the initial angle of the elbow was fixed on 10° and the final angles were changed. The alteration of the desired angles was between 30° and 150° with steps of 10°. The results for rise time, settling time, maximum overshoot and steady state error for comparing the four controllers are shown in Figures 7 to 10.

The next case was studied while the step size between the initial and desired angles was constant (70°) and the initial angles varied

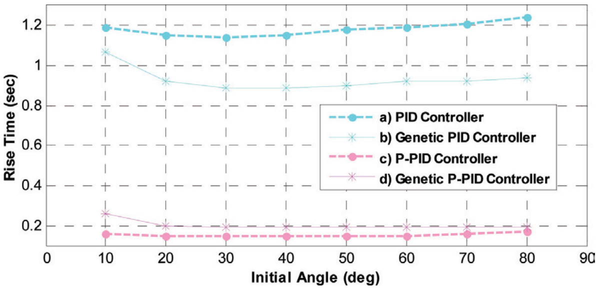

Figure 11: Comparison of the rise time for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same step between the initial and desired angle of 70° and various initial angles (10°–80°)

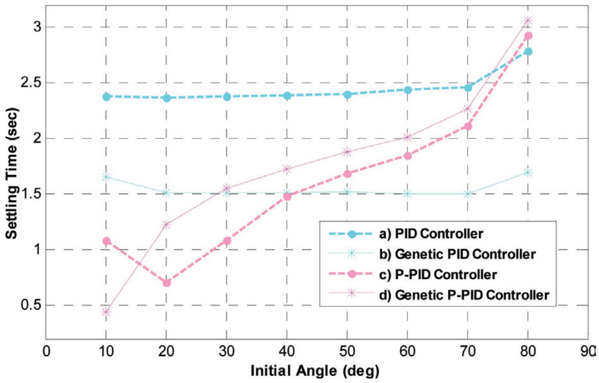

Figure 12: Comparison of the settling time for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same step between the initial and desired angle of 70° and various initial angles (10°–80°)

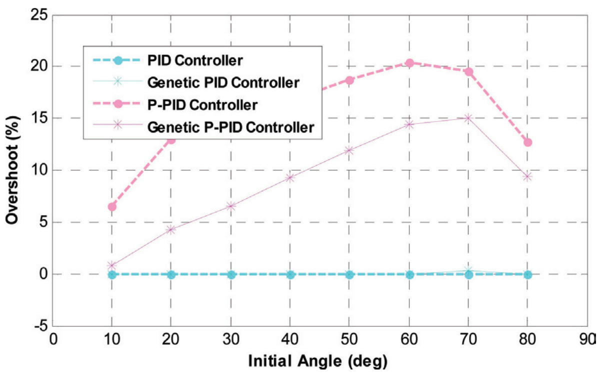

Figure 13: Comparison of the overshoot for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers; with the same step between the initial and desired angle of 70° and various initial angles (10°–80°)

between 10° and 80°. Figures 11 to 13 show the results for the rise time, settling time and maximum overshoot for comparing the four controllers. The steady state error was zero for all the controllers in this condition.

The last case was with a different maximum force for the muscle. The default was 900 N, and we also considered 700 N and 1100 N, as the biceps maximum force to check the robustness of the controllers.

The steady state error was zero in this condition too for all the four controllers. The changes in the rise time, settling time and overshoot are shown in Figures 14 to 16.

Discussion

A feedback PID controller and a feedforward-feedback P-PID controller were developed for a nonlinear musculoskeletal arm model with two muscles, one joint and one degree of freedom in sagittal plane. In feedforward-feedback controller, the feedback part brought the elbow angle to the desired range, and the feedforward part held it on the desired angle. The objective of our study was to improve the controllers' performance using GA for the PID and P-PID tuning.

We first tuned a feedback PID controller by trial and error and then optimized its coefficients by GA that made its performance much better. Then, a combination of feedforward and feedback, as a P-PID, was tuned and optimized by GA. The genetic P-PID performed somewhat better than the P-PID tuned by trial and error. As mentioned earlier, the desired angle that the controllers were tuned based on was 80° while the initial angle was 10°.

The criteria that we checked to determine the performance and robustness of the controllers were the rise time and settling time for the system speed, and maximum overshoot and steady state error as the accuracy of the system. These criteria were measured in three conditions.

The first condition was when the initial angle of the elbow joint was fixed at 10° and the desired angles were altered between 30° and 150° with steps of 10°. In comparing PID, ge-

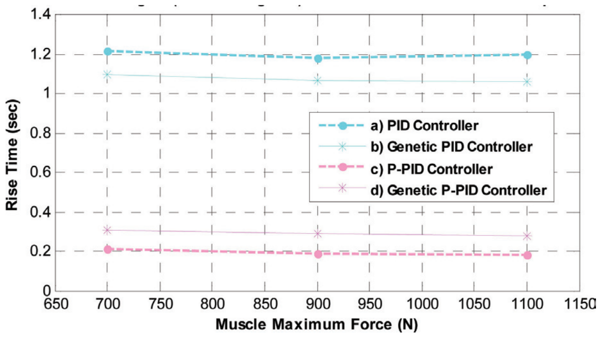

Figure 14: Comparison of the rise time for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers with the same initial and desired angle (10° and 80°, respectively) and various maximum forces for biceps (700–1100 N)

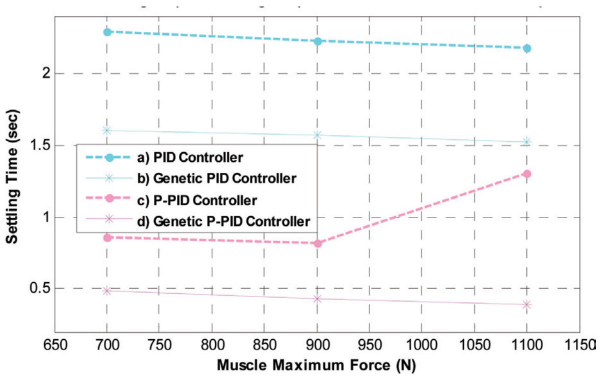

Figure 15: Comparison of the settling time for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers with the same initial and desired angle (10° and 80°, respectively) and various maximum forces for biceps (700–1100 N)

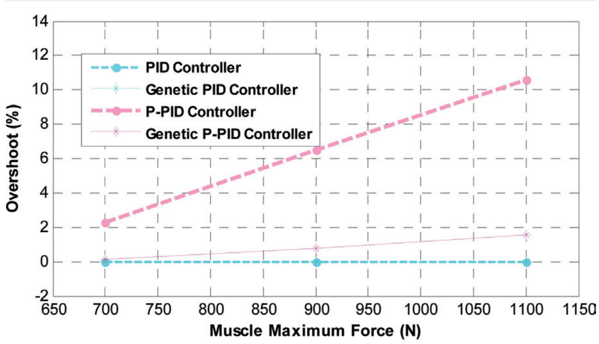

Figure 16: Comparison of the overshoot for a) PID, b) genetic-PID, c) P-PID and d) genetic P-PID controllers with the same initial and desired angle (10° and 80°, respectively) and various maximum forces for biceps (700–1100 N)

netic PID, P-PID and genetic P-PID, in this case, the rise time was much less in both P-PIDs than the PIDs. The settling time in both P-PIDs was smaller than 80° with 10° more or less than that, but not necessarily at other angles. In general, the amounts of settling time in four controllers were not so different. The amounts of overshoot were really better (less) in PIDs and in P-PIDs but were quite large, especially for desired angles >80°. The steady state error was almost nothing for the controllers and exactly nothing for genetic PID controller.

The next condition for checking the robustness was with a fixed step size (70°) and various initial angles. In this case, the rise time in P-PIDs was totally less than PIDs (near 0.2 sec for genetic P-PID and approximately 0.9 sec for genetic PID). Variation in settling time in P-PIDs was rather large (from 0.5 to 3 sec) but in genetic PID it was quite constant at 1.5 sec. The overshoot was zero for PIDs but not for P-PIDs. The steady state error was zero for all the four controllers.

Finally, we checked the performance of the controllers with two different muscle maximum forces beside its default value. In this condition, the rise time was about 0.2 sec for P-PID, about 0.3 for genetic P-PID, almost 1.1 in genetic PID and near 1.2 sec for PID controller. Also, the settling time was great for genetic P-PID (about 0.4 sec), average (1 sec) for P-PID, and almost 1.6 and 2.2 sec for genetic PID and PID, respectively. As the previous conditions, the overshoot was large for P-PID, much better for genetic P-PID and zero for PIDs. And again, the steady state error was zero for all studied controllers.

In conclusion, the combination of feedforward-feedback controller optimized by GA has the best results in general. For comparison of the amount of rise time, we can say that P-PID tuned by trial and error is less but the overshoot in genetic P-PID is much better. Of course, the overshoot in PID feedback controller especially the one tuned by GA is absolutely zero. So, the feedforward-feedback controller causes the system to speed up. However, it brings some overshoot to the system response.

References

1. Blana D, Chadwick EK, Bogert A, Kirsch RF. Feedback Control for a High Level Upper Extremity Neuroprosthesis. *Journal of Biomechanics* 2007;**40**(S2).
2. Kurosawa K, Futami R, Watanabe T, Hoshimiya N. Joint Angle Control by FES Using a Feedback Error Learning Controller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 2005;**13**:359-71.
3. Blana D, Kirsch RF, Chadwick EK. Combined Feedforward and Feedback Control of a Redundant, Nonlinear, Dynamic Musculoskeletal System. *International Federation for Medical and Biological Engineering* 2009;**47**:533-42.
4. Lynch CL, Popovic MR. Closed-Loop Control for FES: Past Work and Future Directions. Proceedings of 10th Annual Conference of the International FES Society: Canada. 2005;47-9.
5. Lynch CL, Popovic MR. Functional Electrical Stimulation: Closed-loop Control of Induced Muscle Contractions. *IEEE Control Systems Magazine* 2008;**28**:40-50.
6. Ferrarin M, Palazzo F, Riener R, Quintern J. Model-Based Control of FES-Induced Single Joint Movements. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 2001; **9**:245-57.
7. Crago PE, Lan N, Veltink PH, Abbas JJ, Kantor C. New Control Strategies for Neuroprosthetic Systems. *Journal of Rehabilitation Research and Development* 1996;**33**:158-72.
8. Reiss J, Abbas JJ. Adaptive Neural Network Control of Cyclic Movements Using Functional Neuromuscular Stimulation. *IEEE Transactions on Rehabilitation Engineering*. 2000;**8**:42-52.
9. Zatsiorsky VM. Kinetics of Human Motion. New Zealand: Human Kinetics Champaign, IL,; **2002** 265-350
10. Zhang D, Ang WT. Tremor Suppression of Elbow Joint via Functional Electrical Stimulation: A Simulation Study. *Proceeding of the 2006 IEEE, International Conference on Automation Science and Engineering:China* 2006;182-187.
11. Maleki A, Shafaei R. Musculoskeletal Model of Arm for FES Research Studies. Proceeding of 4thCario International Biomedical Conference: Egypt. 2008 Dec; 1-4.
12. Riener R, Fuhr T. Patient-Driven Control of FES-Supported Standing Up: A Simulation Study. *IEEE Transactions Rehabilitation Engineering* 1998;**6**:113-124.
13. Bin Mohamed Ibrahim S. The PID Controller Design Using Genetic Algorithm. Bachelor of Engineering Dissertation. Faculty of Engineering and Surveying: University of Southern Queensland. **2005**.
14. Ali MO, Koh SP, Chong KH, Tiong SK, Obaid ZA. Genetic Algorithm Tuning Based PID Controller for Liquid-Level Tank System. Proceedings of the International Conference on Man-Machine Systems (ICoMMS): Malaysia. 2009 Oct.
15. Jagodnik KM, van den Bogert AJ. Optimization and Evaluation of a Proportional Derivative Controller for Planar Arm Movement. *Journal of Biomechanics* 2010;**43**:1086-1091.