

MOTOR UNITS BASED HIERARCHICAL GENETIC ALGORITHM FOR PREDICTION OF MUSCLE FORCE ACTIVATION PATTERNS

Hristo T. Aladjov, Rositsa T. Raikova

Centre of Biomedical Engineering, Bulgarian Academy of Sciences
Acad.G.Bonchev str., Bl.105, 1113 Sofia, Bulgaria
e-mail: aladjov@bgcict.acad.bg, rosir@dir.bg

Keywords: Motor Control, Genetic Algorithm, Motor Unit, Muscle Force, Optimization, Elbow, Reaction

Abstract. The aim of the present paper is to propose a hierarchical genetic algorithm (HGA) for modelling and investigating human limbs motions. HGA is capable of modelling natural hierarchy in brain control mechanisms without restriction on the choice of the effector units model. Each solution of a particular motor task is uniquely represented with the firing rates of motor units (MUs), although it is obtained by satisfaction of multiple goals related to angle displacement, sum of the muscle forces, fatigue, etc. Making the model precise, considering physiological properties of the MUs and muscles, kinematic and dynamic characteristics of the links, increases the computational complexity of the algorithm and deteriorates its performance. To overcome this, new genetic operations and strategies for handling multiple time-dependent goal functions are introduced. The relevance of some machine learning techniques for modelling natural control mechanisms and for performance improvement are also discussed.

1. Introduction

The motion capabilities of leaving beings are formed during their ontogenetic and phylogenetic evolution. This is a continuous process of adaptation related with finding optimal motion strategies and their memorization. Trying to develop models of muscle – joint – bone complexes and their control, one always has to do many simplifications and to cope with our knowledge incompleteness about the living systems. However, it seems reasonable to keep exactly their evolving nature, because this is the property that has made them what they are. Perhaps the most suitable multiobjective optimization and search technique for this purpose is Genetic Algorithm (GA)

(K.Man et al., 1999; M.Vose, 1999). Unlike other optimization techniques (for review see Raikova, 1999), GA can be controlled and monitored during the process of producing solutions.

In order to understand better why a particular muscle is activated during given movement, it is more suitable to provide goals as control input of the algorithm, and then to use experimentally based control signals (such as EMG). The second approach (used in S.Jonic et al., 1999; Y.Koike, M.Kawato, 1995; H.Savelberg, W.Herzog, 1997) can give us only mapping, without helping us to explain the reasons that provoke these control stimuli.

2. Methods

2.1 Hierarchical genetic algorithm

Genetic algorithms represent the potential solutions of a given problem as competing individuals during genetic evolution. These solutions are obtained for different values of design variables regarded as genes of those individuals. During each step of the algorithm (see Fig. 1) a subsequent generation is created by means of genetic operations (mutation and crossover). In the process of evolution the “better” chromosomes generate a larger number of offspring and thus they have higher chances of survival. In HGA, besides the

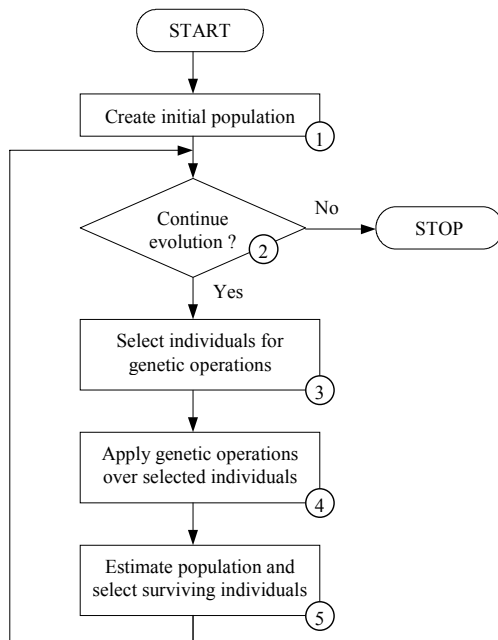


Figure 1: Genetic algorithm flowchart

regular genes, there exist also control genes that activate or inhibit a given subset of regular genes.

2.2 Elbow model

For illustration purposes elbow joint movements are investigated. The model (see fig. 2) considers three flexor (m. biceps brachii - BIC, m. brachialis - BRA, m. brachioradialis - BRD), and two extensor (m. triceps brachii TRI, m. anconeus ANC) muscles.

Each of these muscles consists of four different

types of MUs: I - fast twitch-fatigue resistant (FR); II -

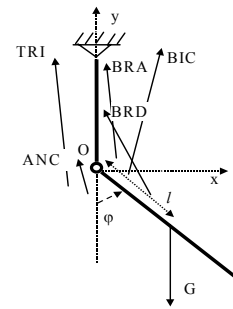


Figure 2: Elbow model

fast twitch-fatiguable (FF); III - intermediate (IM); IV - slow twitch (S) (see Loeb, G.E., 1987). The number of MUs of a muscle is calculated using its physiological cross section area and maximal force.

Elbow joint moment is calculated by:

$$\sum_{i=1}^5 d_i F_i = I_{zz} \ddot{\varphi} + Gl \sin(\varphi) = M_{ext}, \quad (1)$$

where F_i is the i -th muscle force, d_i is the force arm of the i -th muscle, $\ddot{\varphi}$ is the angular acceleration, G is the gravity force of the hand and forearm, l is the distance from O to the application point of G , and M_{ext} denotes the total external moment in the joint. The aim is to calculate individual muscle forces, when external moment of the joint M_{ext} is given. Since this problem is indeterminate one, optimization methods have to be used for its solving.

3. Implementation

3.1 Problem encoding

The main task that must be solved in order to apply GA is how the set of effector units and abstract goals to be represented (encoded) in terms of genetic algorithm theory. First, the design variables must be determined. In living creatures a set of muscle fibers is activated by a moto-neuron. As a result these muscle fibers contract and develop force. Since MUs are fired by electrical stimuli it can be concluded that the natural design variables for GA are the moments of

these activations. Besides information about activation moments, physiological properties of the MUs (such as contraction time, half relaxation time, lead-time, etc.) must be also encoded in the gene. In the present model these properties are used for calculation of the mechanical responses of the MUs (twitches), but they can be also a subject of optimization when trying to model species evolution.

Unlike classical definition of HGA where genes have fixed size, here the length of the genes corresponds to the number of activation moments of the MUs during the whole motion. This corresponds exactly to the “natural design variables”, but hampers the definition of the genetic operations.

The muscle properties are encoded in genes of higher hierarchy level. These genes are composed of the genes corresponding to the muscle’s MUs and a set of genes that encodes physiological properties of the muscle as whole. The next level of hierarchy comprises motor centers. These centers control the set of the muscles and have no exact hierarchical order because they control different aspects of muscle behavior, for example conditioned and unconditioned reflexes.

In natural systems the process of learning and optimizing a new motor task is related to modification of existing motor strategies for similar movements. The idea of genetic operations is to accomplish just those modifications. In the classical definition of HGA there exist two types of operations: mutations and crossovers. In the current study their definition is more complex because of the varying length of the genes.

The idea of mutation operations is to select a particular bit of information (or bits in multipoint mutations) and to alter it. In the context of the current study this can be interpreted as adding or removing stimulation impulse or shifting it back or forward in time.

By analogy with nature, the crossover operations combine the genes of two individual’s chromosomes

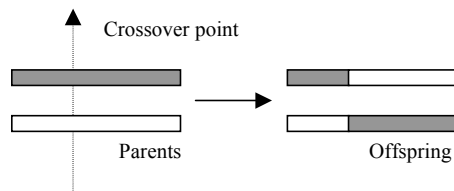


Figure 3: Classical crossover operation

(see Fig.3). The problem that arises when genes are of variable length is that they can not be aligned by sizes and hence divided in the same proportion. Instead of alignment by size, here alignment in time (see Fig. 4) can be used.

The last step in problem encoding is how to

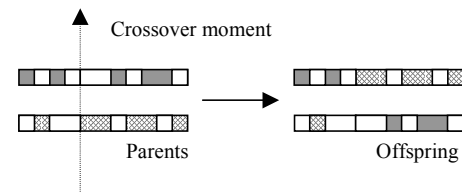


Figure 4: Temporal crossover operation. Filled boxes denotes presence of activation stimuli in that interval, blank an absence

estimate obtained solution and compare it with the desired one. There exist two main problems. The first one is how to estimate more than one goal simultaneously and the second is how to perform this estimation not for a moment but for a time interval. The first problem is often overcome by composing an overall fitness function as a weighted sum of the separate goal estimations, or by using pareto-optimal fitness function (C.Foneska, P.Fleming 1995), where each individual is compared with the rest of the population and its relative “goodness” is evaluated. Both of these approaches are implemented in the program realization. Averaging the errors of the obtained solution for discrete moments from the whole movement time can solve the problem with comparison of time curves. Another approach is to use some transformation of the goal and result functions and to compare parameters of the transformed functions.

3.2 Program realization

A software environment for motor control simulation was developed using HGA. It allows for an interactive definition of the investigated system (muscles, joints, shapes of motor unit twitches, movement goals, etc.) and its properties, as well as monitoring and control of the program execution. These properties can be modified at run time, during the simulation and the changes take effect immediately after current step. This feature can be used for simulating external events and monitoring the response of the investigated system. Such events could be a sudden load of the segments of the body, muscle injuries simulated by removing some of the muscle’s

MUs, changes in goals of the movement and so on.

The HGA is implemented as a standalone Windows application, but this does not encapsulate the project for further development since all the main components are easily exportable and have self-contained functionality. The current state of the project and a demonstration release of the package can be found on the web address: <http://www.motco.dir.bg/>.

4. Results from investigation of elbow flexion motions

The software had been used for prediction of muscle force activation patterns during fast and slow elbow flexion and extension motions. The results show that the used approach predicts well experimental observations such as triphasic-behavior and antagonist co-contraction (for more details see R.Raikova, H.Aladjov, 2001).

In contrast with frequently used objective criteria based on minimization of muscle stresses raised to the

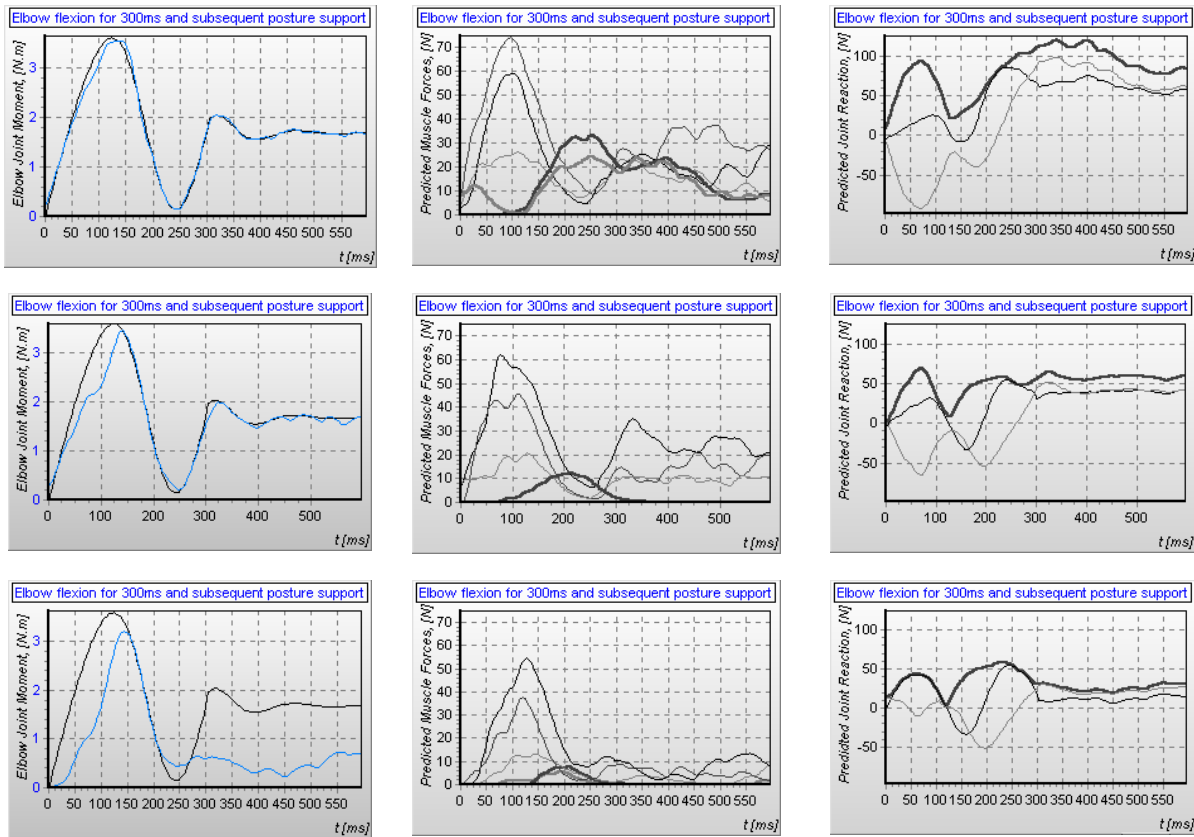


Figure 5: Elbow flexion from 0° to 150° for 300 ms and subsequent posture support. The three rows of the chart are obtained using different approaches for evaluating overall goal satisfaction. First row: objective function is a weighted sum with weights of 1 for the goal Minimal Joint Reaction (MJR) and 1000 for the Joint Angle (JA). Second row: objective function is a weighted sum with MJR weight value 1 and JA goal weight value 100. In the third row Pareto-optimal approach is used. In the first column results (gray curves) and joint moment goals (black curves) are shown, hence there the satisfaction of the JA criterion is compared. The predicted muscle forces are shown in the second column: - BIC (black thin line); - BRA (dark gray thin line) - BRD (light gray thin line) - TRI (dark gray bold line) - ANC (light gray thin line). Predicted joint reactions are shown in the last column: - R (dark gray bold line); horizontal projection: - Rx (black thin line); vertical projection: - Ry (light gray thin line)

power of n here results are obtained using elbow joint reaction (A.Seireg, R.Arvikar, 1973) as one of the goal functions. Three simulations are performed for elbow flexion from 0° to 150° for 300ms and subsequent posture support with different goal weights and estimation mechanisms (see fig.5). The smaller the relative weight of minimal joint reaction criterion as, the lower the value of the predicted joint reaction and muscle forces. However, in case of equal weights (the third row of fig.5) HGA is unable to satisfy joint moment goal. Despite of those differences triphasic-behavior and antagonist co-contraction is predicted in all three cases. It seems that these phenomena are function of the accomplished movement and muscle's model, but not to the optimization criteria.

5. Discussion

Less then the real number of MUs are used in the performed simulation. Simulations with the real number of MUs are also possible, but a compromise between model precision and computational load of the algorithm must be made. The computational load increases because of the resources (time and memory) needed for calculation of the developed forces from MUs using their specific twitch profiles. As far as this model is rather complex (it accounts for fatigue, i.e. the prehistory of the MU activation, it also ignores stimuli that come during the absolute refraction period of the MU and so on), each event that is needed for calculation of MUs force slows down the execution of the algorithm. The most frequent reason for calculating the MU's force output is the estimation of the goal functions. Hence an improvement can be achieved by decreasing the number of estimation points, but in such a way that their representation quality remains approximately the same.

The second factor that deteriorates the computational speed is the duration of the processed movement. The longer is the movement, the largest the amount of stimuli that must be considered in the history of the movement.

To decrease the influence of those factors and to improve performance some new techniques are proposed. It must be noted that cache is used for all calculations. Its purpose is to prevent from over again calculation of unchanged muscle forces.

The first proposed technique for decreasing the computational load is to reduce the number of estimation points of the goal functions. This is

accomplished by a *variable precision* of the estimation. In the beginning the goals are monitored in smaller number of checkpoints and the estimation of the difference between obtained and desired value of the goal is rougher. At each step of the algorithm the precision increases with some step. The additional effect of this is a shift of the estimation points that decreases the impact of estimation discreteness.

Another technique that exploits the same idea is to *oscillate estimation points*, but to keep their number relatively small.

The idea of the *moving frame* approach is to modify and estimate a sub-time interval that moves with some step during the execution. This technique combines the effect of decreased number of estimation points with caching, and with lower error due to averaging the estimations for an interval.

The above described techniques for improving performance of the algorithm can be summarized as methods for reducing estimation points of the goal functions thus reducing the number of moments in which the MU's forces are evaluated. A new genetic operation was also introduced, called "*common synchronous mutation*" that can substantially increase algorithm convergence. The basic idea of this operation is that when a particular goal is optimized with slight or no change in the estimations with respect to other goals, simultaneous modifications of the different part of the chromosome must be accomplished. In particular, when one of the goals is to trace given joint angle displacement (JAD) with some precision and the other is to achieve minimal sum of all muscle forces, common synchronous mutation can simultaneously decrease the activation of some antagonist muscles in one and the same degree. This decreases the sum of forces without changing the estimation with respect to JAD.

Method	Comp. time	Fitness
HGA	02:07:13	0.07156
Cache	01:19:11	0.07156
Variable precision	00:25:37	0.06875
Moving frames	00:23:27	0.06155
Oscillate estimation points	00:24:20	0.06875
Synchronous mutation	00:39:02	0.03636

Table 1: Performance comparison.

A comparison of the proposed improvement techniques is presented in table 1. These results are obtained for 10000 steps, using one and the same initial solutions. It was not a surprise that moving

frames approach achieves the highest speed up (see fig. 6), since it combines the advantages of the smaller number of estimation points, caching and reduced error from averaging. The proposed new

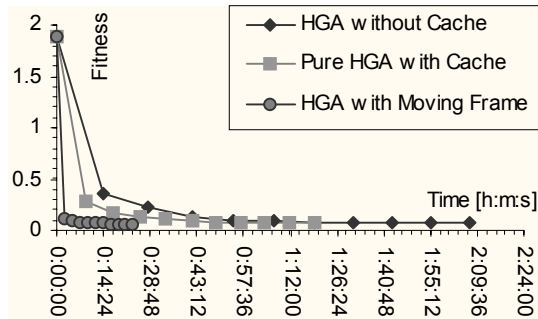


Figure 6: Convergence comparison

genetic operation improves accuracy and can be used in combination with any other technique.

Other non-investigated techniques could be applied in order to boost the algorithm's performance. Such one could be a changing genetic operations probability with respect to the history of their application. The probability of the operation that frequently succeeds in improving the fitness of the individuals can be increased and vice versa.

Only the search aspect of the learning process is discussed in this paper. The other part of learning process - memorization - can be represented by loading initial solutions from set of suitable solutions obtained by solving similar tasks. Neural network or some other classification technique can be used for measuring similarity.

6. Conclusions

The presented HGA realization fits well into the problems of motor control and generates results that are in good agreement with experimental data. The performed simulations show that the algorithm is convergent and robust, producing principally similar solutions during different runs despite its non deterministic nature. It allows for using precise models of MUs, muscles and their hierarchical control mechanisms thus helping us to understand better the problems faced by natural systems. The proposed software implementation provides user-friendly interactive tools for composing different models and for investigating the solution sensitivity to the model parameters. The program realization

provides a set of flexible, interface-independent, event-driven classes that can be easily exported as standalone components for further development.

References

- A.Seireg, R.Arvidar, *A mathematical model for evaluation of forces in lower extremities of the musculo - skeletal systems*. Journal of Biomechanics 6, 313-326, 1973.
- C.Fonesca, P.Fleming, *An overview of evolutionary algorithms in multiobjective optimization*. Evolutionary Computation 3, 1 (Spring) 1-16, 1995.
- G.Loeb, *Hard lessons in motor control from the mammalian spinal cord*. Trends in Neuroscience 10, 108-113, 1987.
- H.Savelberg, W.Herzog, *Prediction of dynamic tendon forces from electromyographic signals: an artificial neural network approach*. Journal of Neuroscience Methods 78, 65-74, 1997.
- K.Man, K.Tang, S.Kwong, *Genetic Algorithms: Concepts and Designs*, London: Springer - Verlag, 1999.
- M.Vose, *The Simple Genetic Algorithm: Foundations and Theory*, A Bradford Book, London: MIT Press, 1999.
- R.Raikova, *About weight factors in the non-linear objective functions used for solving indeterminate problems in biomechanics*. Journal of Biomechanics 32, 689-694. 1999.
- R.Raikova, H.Aladjov, *Hierarchical genetic algorithm versus optimization approach-investigation of elbow flexion and extesion movments*. Journal of Biomechanics (proposed for publication).
- S.Jonic, T.Jankovic, V.Gajic, D.Popovic, *Three machine learning techniques for automatic determination of rules to control locomotion*. IEEE Transactions on Biomedical Engineering 46, 300-310. 1999.
- Y.Koike, M.Kawato, *Estimation of dynamic joint torques and trajectory formation from surface electromyography signals using a neural network model*. Biological Cybernetics 73, 291-300. 1995.