International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-3 Issue-5, November 2013

GA-SVM and MLP-BBO to Estimate Robot Manipulator Joint Angles

Amel Serrat, Mohamed Benyettou

***Abstract*—The kinematic of serial manipulators comprises the study of the relations between joint variables and Cartesian variables. We distinguish two problems, commonly referred to as the direct and inverse kinematic problems. The former reduces matrix multiplications, and poses no major problem. The inverse kinematic problems, however, is more challenging, for it involves intensive variable-elimination and nonlinear-equation solving. In this work, we have used Support Vector Machine with Genetic Algorithm and and Multi Layer Perceptron (MLP) with Biogeography-Based Optimization(BBO) to solve the inverse problem on a manipulator arm, to determine its various articulations. The results of simulation are presented to show the validity of approaches suggested above.**

***Index Terms*—Support Vector Machine, Genetic Algorithm, BBO algorithm, MLP, inverse kinematic, minimally invasive surgery.**

I. INTRODUCTION

Highlight a In robotics, to determine the various articulations of arm is significant and essential, which is even used for prediction of Protein's structures [1]-[6]. The problem consists in finding the parameters, which bring the final point to a wished situation, being given a configuration of the arm with chains series and the final point. The problem is so difficult because it requires resolution of a system of nonlinear equations, and the difficulty increases with the number of bonds in the chained structure. There is not a general analytical method to circumvent this problem.

Numerous solution strategies have been proposed for the Inverse Kinematic Problem, such as distance matrix completion [7], characteristic polynomial, dyalitic elimination, genetic programming, intelligent algorithm, and wavelet networks [2]. We present a new approach, Artificial Immune System, to solve this problem without constraint or restrictions on structure of arm.

This work is organized as follows. Some concepts of the robotic are given first, after we construct the Artificial Immune System and finally we discuss the results of the application.

II. ROBOT MANIPULATOR

All A robot is a combined mechanical, electronic, and computer system that follows a simple cycle of commands and task execution for operation. First, the computer learns environmental information from its sensors. Based on this information and the task to be accomplished, computer algorithms calculate appropriate commands for the motors. These commands are sent to the mechanical system, which executes the task, and the cycle repeats, all actions of the robot have to be continually monitored to correct deviations from the planned trajectory.

The range of motion of each manipulator is called working space. The manipulator is normally connected to a base (floor, ceiling, operating table, etc) and composed of a succession of joints and links (appendages). The instrument with which the robot performs the desired task is attached to the last link of the arm and is referred to as the end-effector. An end-effector can be a needle, grasper, scalpel,…etc.

Our model of robot is AESOP with 6 joints angles (Q1, Q2, Q3, …,Q6), and we must find them for a given position in Cartesian space[8]. Here is the pseudo code of it:

```
  Input: position to be reached (M)
  Output: values of the parameters of the
articulations
     Q= vector of model parameters
              Q = (Q1, Q2, Q3, …,Q6)
  Find Q subject to:
     f(Q)-M=0
   which brings back us to min ||f(Q)-M||.
```

III. GENETIC ALGORITHM (GA)

Genetic Algorithm evolves a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved.

Each individual is called chromosome, and is composed of a predetermined number of genes. The quality of each rule is measured by a fitness function as the quantitative representation of each rule's adaptation to a certain environment. The procedure starts from an initial population of randomly generated individuals. Then the population is evolved for a number of generations while gradually improving the qualities of the individuals in the sense of increasing the fitness value as the measure of quality. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, i.e. selection, crossover and mutation. The algorithm flow is presented in Fig. 1. Determination of the following factors has the crucial impact on the efficiency of the algorithm: selection of fitness function, representation of individuals and the values of GA parameters (crossover and mutation rate, size of population, threshold of fitness value).

Manuscript Received November 2013.
Amel SERRAT, LAMOSI, dept. of Computer Sciences, University Of Sciences and Technologies Oran, 31000, Algeria.
Mohamed Benyettou, LAMOSI, dept. of Computer Sciences, University Of Sciences and Technologies Oran, 31000, Algeria.

Determination of these factors usually depends on the application.

- Generate random population of n chromosomes (suitable **solutions** for the problem.
- Evaluate the fitness f(x) of each chromosome x in the population.
- Create a new population by repeating the following steps until the new population is complete.
  - Select the best chromosome or chromosomes to be carried over to the next generation.
  - Select two parents constitute stochastic evolutionary techniques whose research methods model some natural phenomena [8].

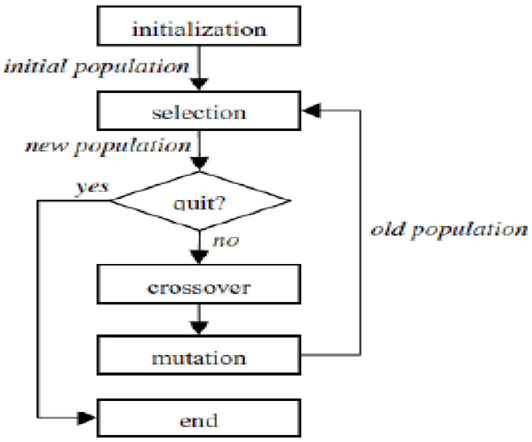

Fig. 1: Genetic Algorithm Flowchart

IV. SUPPORT VECTOR MACHINE

Consider the following problem: we are given a data set $g = \{(x_i, y_i)\}_{i=1}^{N}$ obtained by sampling, with noise, some unknown function *f(x)* and we are asked to recover the function f , or an approximation of it, that has at most ε deviation from the actually obtained targets y_i for all the training data g , and at the same time is as flat as possible. In other words, we do not care about errors as long as they are less than, but will not accept any deviation larger than this [9].

$$|y - f(x)|_{\varepsilon} := \max\{0, |y - f(x)| - \varepsilon\}. \tag{1}$$

A. Linear Case

We have:

$$f(x) = \langle w, x\rangle + b, \qquad b \in \Re. \tag{2}$$

Where $\langle \cdot,\cdot \rangle$ denotes the dot product in the space of the input patterns. Flatness in the case of (2) means that one seeks a small ***w***. One way to ensure this is to minimize the norm. We can write this problem as a convex optimization problem [10].

$$\min \quad \frac{1}{2}\|w\|^2 . \tag{3}$$

$$\text{where} \quad \begin{cases} y_i - \langle w, x_i\rangle - b & \in \varepsilon \\ \langle w, x_i\rangle + b - y_i & \in \varepsilon \end{cases} \quad . \tag{4}$$

The tacit assumption in (2) was that such a function actually exists that approximates all pairs (x_i, y_i) that the convex optimization problem is feasible.

Sometimes, however, this may not be the case, or we also may want to allow for some errors. One can introduce slack variables ζ_i to cope with otherwise infeasible constraints of the optimization problem (2).

We define:

$$\zeta_i = \max\{0, |y_i - f(x_i)| - \varepsilon\}. \tag{5}$$

$$\begin{cases} \zeta_i^+ = \max\{0, y_i - f(x_i) - \varepsilon\} \\ \zeta_i^- = \max\{0, f(x_i) - y_i - \varepsilon\} \end{cases}. \tag{6}$$

$$\zeta_i = \max\{\zeta_i^+, \zeta_i^-\}. \tag{7}$$

Hence we arrive at the formulation stated in [11].

$$\min \quad \frac{1}{2}\|w\|^2 + C\sum_i (\zeta_i^+ + \zeta_i^-) \tag{8}$$

$$\text{with} \quad \begin{cases} y_i - \langle w, x_i\rangle - b - \varepsilon & \le \zeta_i^+ \\ \langle w, x_i\rangle - y_i + b - \varepsilon & \le \zeta_i^- \\ \zeta_i^+, \zeta_i^- \ge 0 \end{cases} . \tag{9}$$

The constant ***C* > *0*** determines the trade-off between the flatness of ***f*** and the amount up to which ε deviations larger than are tolerated.

In most cases the optimization problem (9) can be solved more easily in its dual formulation. Moreover, the dual formulation provides the key for extending SV machine to nonlinear functions. Hence we will use a standard dualization method utilizing Lagrange multipliers, as described [12,13].

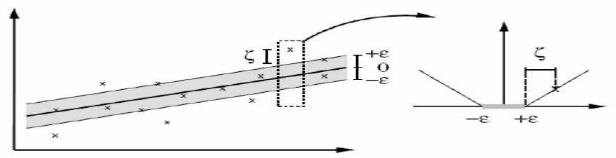

Fig. 2: The soft margin loss setting.

B. Dual Problem and Quadratic program

The idea is to build a function of Lagrange by the objective function and its constraints by introducing a dual set of variables. It can be shown that this function has a saddle point with respect to the primal and dual variables at the solution. For details see e.g. [14]. We proceed as follow:

$$\begin{aligned} L = {} & \frac{1}{2}\|w\|^2 + C\sum_i \alpha_i(\varepsilon + \zeta_i^+ - y_i + \langle w, x_i\rangle + b) \\ & - \sum_i \alpha_i^*(\varepsilon + \zeta_i^- + y_i - \langle w, x_i\rangle - b) - \sum_i (\eta_i \zeta_i^+ + \eta_i^* \zeta_i^-) \end{aligned} . \tag{10}$$

$$\text{with} \quad \alpha_i, \alpha_i^*, \eta_i, \eta_i^* \quad \ge 0.$$

And $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$ are the Lagrange Multipliers.

The Lagrangian has to be minimized with respect to ***w, b*** and maximized with respect to ***α≥ 0***.

At point of optimality we have:

$$\partial_b L = \sum(\alpha_i - \alpha_i^*) = 0. \tag{11}$$

$$\partial_w L = w - \sum(\alpha_i - \alpha_i^*)x_i = 0.$$

$$\partial_{\zeta_i^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 . \tag{12}$$

(13)

Substituting Eq.(11, 12,13) in eq.(10) we get:

$$L = \frac{1}{2}\sum_{i,j}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle. \quad (14)$$

$$\text{With } \begin{cases} \sum(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \quad i \in [0, C] \end{cases}. \quad (15)$$

In deriving (10) we already eliminated the dual variables η_i, η_i^* through condition (Eq.13) which can be reformulated as:

$$\eta_i^* = (C - \alpha_i). \quad (16)$$

Equation (12) can be written as follows:

$$w = \sum(\alpha_i - \alpha_i^*)x_i. \quad (17)$$

Thus;

$$f(x) = \sum(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b. \quad (18)$$

What is called support vectors expansion i.e. ***w*** can be completely described as a linear combination of the training patterns. In a sense, the complexity of a function's representation by SVs is independent of the dimensionality of the input space, and depends only on the number of SVs.

Moreover, note that the complete algorithm can be described in terms of dot products between the data. Even when evaluating ***f (x)*** we need not compute ***w*** explicitly.

C. Computing b

b can be computed by exploiting the so called Krush-Kuhn-Tucker (KKT) conditions [15]. These state that at the point of the solution the product between dual variables and constraints has to vanish.

$$\alpha_i(\varepsilon + \zeta_i^+ - y_i + \langle w, x_i\rangle + b) = 0. \quad (19)$$

And

$$(C - \alpha_i)\zeta_i^+ = 0. \quad (20)$$

$$(C - \alpha_i^*)\zeta_i^- = 0. \quad (21)$$

For points which are apart from the band (ε-insensitive tubes) we have

$$\alpha_i^{(*)} = C. \quad (22)$$

The points inside the band have

$$\alpha_i^{(*)} = 0. \quad (23)$$

The others

$$0 < \alpha_i^{(*)} < C. \quad (24)$$

The latter one has

$$\zeta = 0. \quad (25)$$

Consequently ***b*** calculation as follows [13]:

$$b = y_i - \langle w, x_i\rangle - \varepsilon \quad \text{for} \quad \alpha_i \in [0, C]. \quad (26)$$

$$b = y_i - \langle w, x_i\rangle + \varepsilon \quad \text{for} \quad \alpha_i^* \in [0, C]. \quad (27)$$

From Eq.(16) it follows that only for $|y - f(x)| \geq \varepsilon$ the Lagrange multipliers may be nonzero, or in other words, for all samples inside the ε- insensitive tube, they vanish for $|y - f(x)| < \varepsilon$ the $\alpha_i^{(*)}$ has to be zero such that the KKT conditions are satisfied. Therefore we don't need all the points to define ***w*** .The examples that come with non vanishing coefficients are called Support Vectors.

D. Non Linear Case

This concept can be extended to the case when ***f*** is non linear. A non-linear mapping which maps the input data to a high dimensional space (also called the feature space) is introduced. We can then try to find a linear function in feature space.

Thus we avoid translating the input data to feature space first and then finding their inner products.

The difference in the linear case is that w is no longer given explicitly. Also note that in the nonlinear setting, the optimization problem corresponds to finding the flattest function in feature space, not in input space [16].

V. BIOGEOGRAPHY BASED OPTIMIZATION

It is a method described by Dan Simon in 2008 [17], it projects the Biogeographic study to the optimization which aims at studying the distribution of the species in the biosphere (biological system) [18-20]. It is based on two fundamental concepts Immigration and Emigration of the species between islands, each island is a set of habitats.

Each habitat is defined by [20]:

1) Habitat Suitability Index (HSI) : According to some parameters, habitats are comfortable, advantageous highly habitable, and presents a strong attraction of the other species towards them we say that they have high Habitat Suitability Index(HSI), and if they present a weak attraction we say that they have low Habitat Suitability Index. Analogically with the evolutionary algorithms habitats with High HSI is a good solution and the other is a bad or poor solution. This comfort is defined by the fitness function.
2) Suitability Index Variable (SIV): Or parameters leading to the comfort of the habitat. Biologically this comfort refers the diversity of the vegetation, surface, temperature… etc.
3) The number of species: is the number of Animals that the habitat comprises. Habitats with High HSI have several species, and then the number of species is proportional to HSI.
4) The emigration rate: habitats with high HSI have high emigration rate.
5) The immigration rate: Habitats with high HSI have low immigration rate, because habitats sature and do not accept new species.
6) The largest number of species: Habitats have threshold to control the migration.

BBO algorithm comprises three steps: migration, mutation, elitism; the mutation and elitism are not essentials in BBO and the migration algorithm is as follow:

```
For each habitat
  Select I_i with probability based on λ_i
  If I_i is selected
    For each habitat
```


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

```
      Select I_j  with probability based on  μ_j
      If  I_j is selected
      Randomly select a SIV x from I_i
      Replace a random SIV in I_j with  x
      End
    End
  End
End
```

VI. EXPERIMENTAL RESULTS

Simulation on the arm was made with a processor Pentium 4 given frequency of 1.8 GHz, with HDD of 40 GO, a RAM of 1 GO, and under Matlab 7. Above all, we present the result found with Wavelet network in 2006[21]:

TABLE I. MEAN SQUARE ERROR WITH WAVELET NETWORK

	Errors (×10−3)
Q_1	0.2766
Q_2	0.1446
Q_3	0.0292
Q_4	0.2591
Q_5	0.1984
Q_6	0.1512

We design a genetic algorithm in order to learn the SVM parameters which are C, kernel function and its parameters, and the loss function.

We implemented GA-SVM with these parameters:

TABLE II. GA-SVM PARAMETERS

Parameter	Generation	mutation Probability
Value	300	0.2

We design a BBO algorithm in order to learn the MLP weights.

We implemented MLP-BBO with these parameters:

TABLE III. MLP-BBO PARAMETERS

Parameter	Generation	mutation Probability	β
Value	400	0.001	0.1

TABLE IV. GA-SVM AND MLP-BBO RESULTS

Angles	GA-SVM error $*10^{-4}$	BBO error $*10^{-3}$
$\mathbf{Q_1}$	0.28	0.11
$\mathbf{Q_2}$	0.04	0.06
$\mathbf{Q_3}$	0.14	0.078
$\mathbf{Q_4}$	0.02	0.34
$\mathbf{Q_5}$	0.03	0.021
$\mathbf{Q_6}$	0.001	0.0218

As we see GA-SVM has estimate angles with error better than BBO-MLP and wavelet algorithm. GA has learned the SVM parameters efficiently.BBO also has learned the MLP weights with error of 10^{-3} order.

VII. CONCLUSION

According to the experimentation made on AESOP, we notice that both BBO-MLP and GA-SVM used previously gave satisfying results by comparing it with Wavelet network.

As a prospect, we suggest implementing it in the controllers of the robot.

VIII. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

REFERENCES

1. R. Kolodny, L.Guibas , M. Levitt , P. Koehl.:Inverse kinematics in biology: The protein loop closure problem. Int. J. Robot. Res., vol. 24, no. 2-3, 2005, pp. 151–163.
2. H.Van den Bedem , I.Lotan,J. Latombe , A. M. Deacon: Realspace protein-model completion: an inverse-kinematics approach. Acta Crystallographica, vol. D61, 2005, pp. 2–13.
3. J. M.Porta, L.Ros, F. Thomas.: Inverse Kinematics By Distance Matrix Completion. Proceedings of CK2005, 12th International Workshop on Computational Kinematics Cassino May 4-6, 2005.
4. C. Hua, C. Wei-shan, T. Xie:Wavelet network solution for the inverse kinematics problem in robotic manipulator. Journal of Zhejiang University SCIENCE A, 2006.
5. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1476144/ accessed 17/10/2013.
6. M.Ghodousssi M., Butner S., Wang Y.: Robotic surgery – the transatlantic case. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, 2002.
7. A. Preumont : Théorie Générale des Systèmes Articulés. 2nd Edition , UBR, 2001.
8. P. Gupta and K. Shinde.:Genetic Algorithm Technique Used to Detect Intrusion Detection, Proc. of the 1st International Conference (ACITY 2011),July 2011, pp. 122-131.
9. V.Vapnik .: The Nature of Statistical Learning Theory , Springer, New York,1995.
10. K. P. Bennett. and O.L. Mangasarian.:«Robust linear programming discrimination of two linearly inseparable sets». Optimization Methods and Software,1992, pp 1,23-34.
11. A.J. Smola . and B. Scholkopf .:A Tutorial on Support Vectors regression». NeuroColt2,1998.
12. McCormick G. P.: Nonlinear Programming: Theory, Algorithms, and Applications. John Wiley and Sons, New York,1983.
13. L. Miclet L.: Apprentissage Artificiel : Méthodes et Algorithmes. Eyrolles, Novembre, 2002.
14. H.W. Kuhn and A. W. Tucker,: Nonlinear programming. In Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics, Berkeley, University of California Press. 1951, pp 481-492
15. N.Leonard de Castro, F. J. Zuben.:Learning and Optimization Using the Clonal Selection Principle, IEEE Transaction on Evolutionary Computation, vol. 6. no. 3, 2002.
16. D. Simon;: Biogeography-based optimization. IEEE Trans. on Evolutionary Computation. 12, 6,2008, pp. 702–713.
17. A. Wallace.: The Geographical Distribution of Animal. Adament Media Corporation, 2006 (first published in 1876).
18. C. Darwin . : The origin of species. Gramercy 1995 (first published in 1859).
19. V. Panchal, P.Singh, N. Kaur , H. Kundra.:Biogeography based satellite image classification. International Journal of Computer Science and Information Security, vol. 6, no. 2, 2009, pp. 269-274.

AUTHOR PROFILE

Amel Serrat was born in Oran, Algeria, on April, 1983. She received the B.S. and M.S. degrees from the Department of Computer Engineering, University of Sciences and Technologies Oran, Algeria, in 2006 and 2009. From 2005 to now, she is Member of Technical Staff with LAMOSI Laboratory.

Mohamed Benyettou was born in Algeria. He received his Ph.D. third cycle degree from the Nancy University II, France, in 1985. He is the director of Laboratory of Modeling and Optimization of Industrial Systems LAMOSI. He acquires many awards and appreciations form different parties for excellent teaching and extra scientific research efforts. Also, he was invited to join many respectable scientific organizations. .