

## Comparison Between Genetic Algorithm and Electromagnetism-Like Algorithm for Solving Inverse Kinematics

<sup>1</sup>Issa Ahmed Abed, <sup>1</sup>S.P. Koh, <sup>1</sup>Khairul Salleh Mohamed Sahari,  
<sup>1</sup>S.K. Tiong and <sup>2</sup>David F.W. Yap

<sup>1</sup>College of Engineering, Universiti Tenaga Nasional, Selangor, Malaysia

<sup>2</sup>Faculty of Electronic and Computer Engineering,  
Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

---

**Abstract:** A comparison study between Electromagnetism-Like Algorithm (EM) and Genetic Algorithm (GA) has been presented in this work to solve the Inverse Kinematics (IK) of a four-link planar robot manipulator. The comparison is focused on some points for both algorithms like the accuracy of the results and the speed of convergence. Different target points have been taken to check the performance of each algorithm to solve the IK problem. The results showed that EM algorithm needs less population size and number of generations to get the true solution. There are multiple robot configurations at the goal points and both algorithms are able to find these solutions at each point. Self developed software simulator is used to display some of these solutions at each goal position.

**Key words:** Inverse kinematics • Real coded genetic algorithm • Attraction-repulsion mechanism • Planar manipulator

---

### INTRODUCTION

Nowadays, there are many applications for robots. These robots have non-linear kinematics equations. Their inverse kinematics solution provides the joint angles which are required to attain a particular position of the robot wrist in the robot work space [1]. The mapping from joint space to the end effector space is referred to as Forward Kinematics (FK). Finding the joint angle of the manipulator from end effector position is referred as Inverse Kinematics (IK). The forward kinematics equations can be solved easily, but it is difficult to solve inverse kinematics exactly for high-order degree of freedom.

Many approaches have been proposed to solve inverse kinematics equations. One of these approaches is to use numerical methods [2]. In numerical methods, a good initial guess must be given because these methods are divergence and vulnerable to local optimums [3].

Recently, artificial intelligence methods were applied to solve the inverse kinematics problem. Köker *et al.* [4] have designed a neural network to solve the inverse

kinematics for a three joint robot. The initial and final points have been generated by using cubic polynomial. After that, all the angles that obtained from (x, y, z) coordinates are recorded in a file named as training set of neural network. Based on metaheuristics algorithms, Chandra and Rolland [5] proposed hybrid algorithm based on genetic algorithm and Simulated Annealing (SA) to solve the forward kinematics of the 3RPR parallel manipulator. In this method, both algorithms are hybridized into two hybrid metaheuristic techniques. One of the limitations in this method is long optimization time. Two examples for SCARA and PUMA robots have been taken by Kalra *et al.* to check an evolutionary approach based on real-coded genetic algorithm to get the solution of the multimodal inverse kinematics problem [1].

In the last years, new metaheuristic methods such as EM algorithm have been used to solve the inverse kinematics for robot manipulators. EM algorithm is an optimization algorithm that uses the principle of attraction-repulsion mechanism. EM algorithm has been applied to different problems, such as optimizations problems [6], scheduling problems [7] and

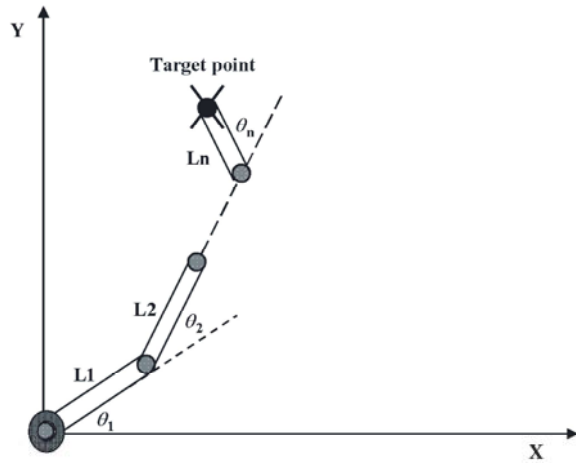


Fig. 1: n links planar robot

so on. Feng *et al.* [8] suggested a method based on electromagnetism-like algorithm and modified Davidon-Fletcher-Powell (DFP) for inverse kinematics. They developed a modified DFP algorithm to fine-tune the approximation results from EM algorithm, at the desired accuracy.

In this paper, a comparison study between EM algorithm and GA has been presented to solve the inverse kinematics for planar robot manipulator. The remainder of the paper is divided as follows. Section 2 is the discussion for problem formulation and the objective function; Section 3 and Section 4 give the explanation for GA and EM algorithm, respectively; Simulation results have been presented in section 5 for both EM algorithm and GA and these results showed the behaviors of each method; finally, the conclusion has been given in Section 6.

**Problem Formulation:** In order to design any job in robotics systems, forward kinematics is required and subsequently, the inverse kinematics. According to the Figure 1, geometric method can be used to find the forward equations for the n DOF planar robot [9] as follows:

$$X_{cur} = L1 * \cos\theta_1 + L2 * \cos(\theta_1 + \theta_2) + \dots + Ln * \cos(\theta_1 + \theta_2 + \dots + \theta_n) \quad (1)$$

$$Y_{cur} = L1 * \sin\theta_1 + L2 * \sin(\theta_1 + \theta_2) + \dots + Ln * \sin(\theta_1 + \theta_2 + \dots + \theta_n) \quad (2)$$

where  $L_n$  denotes the  $n$ th link length,  $\theta_n$  is the  $n$ th joint angle and  $(X_{cur}, Y_{cur})$  is the current point.

The error between the current point  $(X_{cur}, Y_{cur})$  and the target position  $(X_{tp}, Y_{tp})$  of the end effector according to [10] is given as follows:

$$Error = \sqrt{(X_{tp} - X_{cur})^2 + (Y_{tp} - Y_{cur})^2} \quad (3)$$

Thus, the inverse kinematics problem is to search and find at least one solution  $\theta' = \{\theta'_1, \theta'_2, \dots, \theta'_n\}$  that approximates this error to zero. As a result, the problem has been transformed to the minimization of this error.

**Real Valued Genetic Algorithm:** A genetic algorithm is a global optimization algorithm that depends on the concept of biological structures to natural selection and survival of the fittest [11]. Various operators are used in GA as genetic operators which produce the best individuals from an initial random population. Genetic operators have an important role in the convergence of the genetic algorithm. The basic components of a genetic algorithm are as follows:

**Initialization:** In the beginning, the individuals are randomly generated,  $(\theta^1, \theta^2, \dots, \theta^m)$ . These individuals or chromosomes represent the problem variables. The solutions are seeded in the area of search space where the optimal solution is possible to be found [5, 12].

**Evaluation:** In each generation, the parameters of chromosomes are sent to evaluation function to measure the solution value [13].

**Selection:** The chromosomes will be selected for next generation and this selection will be based on the fitness function of the chromosome.

**Elitism:** It is used to ensure that the best chromosome always dominates in the population. This process is to avoid the possibility of losing the better chromosome.

**Crossover:** In this reproduction operation, the offspring can be generated by exchanging the genetic material between the selected parameters. Some methods for crossover have been discussed by many researchers such as arithmetic crossover [14, 15] and single point crossover [16]. In arithmetic crossover, two parents are selected randomly from the pool and linear combinations between these two parents' genes are done to produce the offspring as follows:

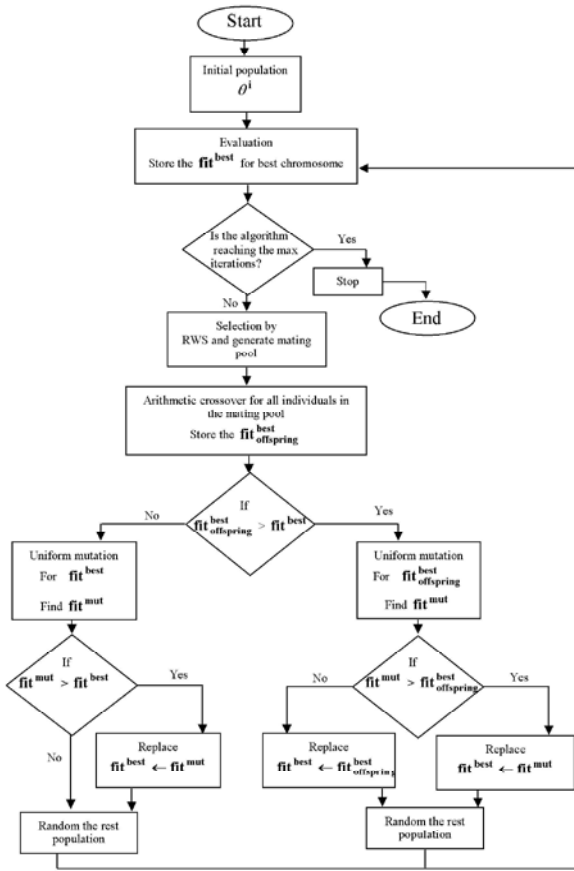


Fig. 2: Proposed GA

$$\theta_i^{gen+1} = \theta_j^{gen} + \alpha(\theta_i^{gen} - \theta_j^{gen}) \quad (4)$$

$$\theta_j^{gen+1} = \theta_i^{gen} + \alpha(\theta_j^{gen} - \theta_i^{gen}) \quad (5)$$

where  $\theta^{gen}$  is the gene of the old generation,  $\theta^{gen+1}$  is the gene of the new generation and  $\alpha$  is a random number with uniform distribution between zero and one.

**Mutation:** It is the operator which provides the variation of individuals during the generations of GA. The mutation is to avoid the local minimum. Uniform mutation [14] is the method that has been used in this paper. In this type of mutation, a random number on interval  $[-s, s]$  is generated and add to the genes of the chromosome which is selected to the mutation operation.

Figure 2 shows the steps of our proposed genetic algorithm. In this flow chart,  $fit^{best}$  is the fitness of the best chromosome.  $fit^{best}_{offspring}$  is the fitness of the best chromosome after crossover.  $fit^{mut}$  is the fitness of the chromosome that produces from mutation.

**Electromagnetism-like Mechanism:** EM is a population-based algorithm. It differs from GA and SA in terms of exchanging the materials between the population members. On the other hand, EM algorithm is like Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) in terms of the influence by all other particles in the population [17]. EM algorithm is a good algorithm to solve the IK regardless to geometry and the DOF for robot [8]. The main procedures for the algorithm are:

**Initialization:** A population with  $m$  points is randomly generated, which is  $n$  dimensional. Each dimension has upper bound and lower bound [18]. After the generation of the samples, objectives function value for each sample is evaluated. Algorithm 1 shows the initializing procedures.

#### Algorithm 1:

For  $i = 1$  To  $m$  do

For  $k = 1$  To  $n$  do

$\lambda \leftarrow \text{Uniform}(0, 1)$

$\theta_k^i \leftarrow l_k + \lambda(u_k - l_k)$

End for

Calculate  $f(\theta_i)$

End for

$\theta^{best} \leftarrow \arg \min \{f(\theta^i), \forall i\}$

where

$m$  = Is the number of sample points (population size).

$n$  = Is the dimension for each sample.

$l_k$  = Lower bound of the  $k$ -th dimension.

$u_k$  = Upper bound of the  $k$ -th dimension.

$f(\theta^i)$  = Is the objective function and it is equal to the error.

**Local Search:** Is the procedure that gathers the local information of each sample point. Algorithm 2 shows the local search procedures [19].

**Algorithm 2:**

Length  $\leftarrow \delta(\max\{u_k - l_k\})$

For  $i = 1$  To  $m$  do

For  $k = 1$  To  $n$  do

Counter  $\leftarrow 1$

DO

$\lambda_1 \leftarrow U(0, 1)$

$y \leftarrow \theta^i$

$\lambda_2 \leftarrow U(0, 1)$

If  $\lambda_1 > 0.5$  then

$y_k \leftarrow y_k + \lambda_2 * \text{Length}$

Else

$y_k \leftarrow y_k - \lambda_2 * \text{Length}$

End if

If  $f(y) < f(\theta)$  then

$\theta^i \leftarrow y$

Counter  $\leftarrow \text{LSITER} - 1$

End if

Counter  $\leftarrow \text{counter} + 1$

Loop while counter  $< \text{LSITER}$

End for

End for

$\theta^{\text{best}} \leftarrow \arg \min \{f(\theta), \forall i\}$

where, LSITER denotes the local search iterations and in this paper, it sets to 10 iterations,  $\delta$  represents the local

search parameter and  $\delta \in [0, 1]$  After the initialization of population, the algorithm starts with local search procedures. First, the maximum feasible step length (Length) is evaluated by using the parameter  $\delta$ . Second, for each  $i$  the initial point  $\theta^i$  is stored in the temporary point  $y$ . Third,  $y$  is moved according to the selected random number coordinate by coordinate. Fourth, if the new point in  $y$  is a better point than  $\theta^i$  within LSITER iterations, then  $\theta^i$  will be replaced by  $y$  and the search for this  $\theta^i$  will finish. Or else, the counter is increased by one and the loop returns again to another iteration for LSITER. Finally,  $\theta^{\text{best}}$  is updated.

**Total Force Calculations:** The first step in the force calculation is the calculation of the charge for each sample point. The charge calculations for each sample point are performed for each generation and it depends on the objective function of this point and the objective function for the best point as shown in the Eq. (6) [17] :

$$q^i = \exp \left\{ -n \frac{f(\theta^i) - f(\theta^{\text{best}})}{\sum_{k=1}^m [f(\theta^k) - f(\theta^{\text{best}})]} \right\}, \forall i \quad (6)$$

where  $f(\theta^i)$  is the objective function value of sample point  $\theta^i$  and  $f(\theta^{\text{best}})$  is objective function of current best solution. Observe that, no sign appeared on the charge of a sample point. In its place, the difference between the objectives functions of two points will decide the direction of the force between them. Hence,

$$F^i = \sum_{j \neq i}^m \left\{ \begin{array}{ll} (\theta^j - \theta^i) \frac{q^i q^j}{\|\theta^j - \theta^i\|^2} & \text{if } f(\theta^j) < f(\theta^i) \quad (\text{Attraction}) \\ (\theta^i - \theta^j) \frac{q^i q^j}{\|\theta^j - \theta^i\|^2} & \text{if } f(\theta^j) \geq f(\theta^i) \quad (\text{Repulsion}) \end{array} \right\}, \forall i \quad (7)$$

where  $F^i$  is the total force exerted on sample point  $\theta^i$ . According to Eq. (7), the point with better objective function attracts the other points. Even so, the point with worse objective function value will repel the others.

**The Movement along the Total Force:** The last step after force calculation is to calculate the movement according to the force. So, the point will move in the direction of the force by random step length. Algorithm 3 shows the movement steps.

**Algorithm 3:**

For i = 1 To m do

If i ≠ best then

$\lambda \leftarrow U(0, 1)$

Norm calculation for the force  $\|F^i\|$

$$F^i \leftarrow \frac{F^i}{\text{norm}}$$

For k = 1 To n do

If  $F_k^i > 0$  then

$$\theta_k^i \leftarrow \theta_k^i + \lambda F_k^i (u_k - \theta_k^i)$$

Else

$$\theta_k^i \leftarrow \theta_k^i + \lambda F_k^i (\theta_k^i - l_k)$$

End if

End for

End if

End for

**Experimental Results:** In order to compare the performance of EM algorithm with GA, simulators were conducted using a four-link planar robot manipulator. The simulators were conducted using a self developed GUI to find the inverse kinematics of the manipulator. The links lengths for planar robot are 20 cm each. The upper and lower bound for joint angles are  $180^\circ$  and  $0^\circ$ , respectively. For the comparison between EM algorithm with GA, three target points have been taken to find the inverse kinematics at these points. The population size for both algorithms is 100 and the maximum number of iterations is 1000. All the tests and the results have been done on Celeron ® CPU 2.2 GHz PC. The algorithms are developed and tested using Visual Basic 2008. Figure 3 shows the convergence error for both algorithms. Figures 4 (a)-(c) and Figures 5 (a)-(c) show the movement of the population to the solution for EM algorithm and GA, respectively.

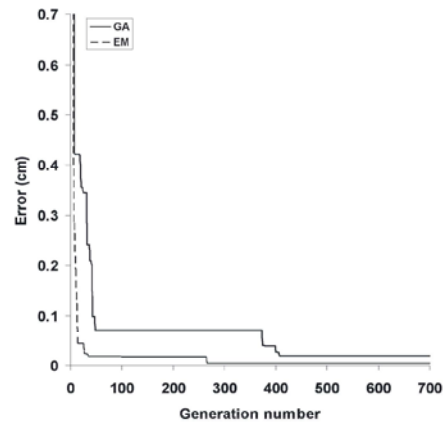
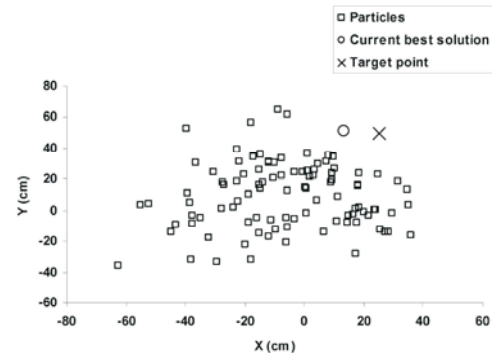
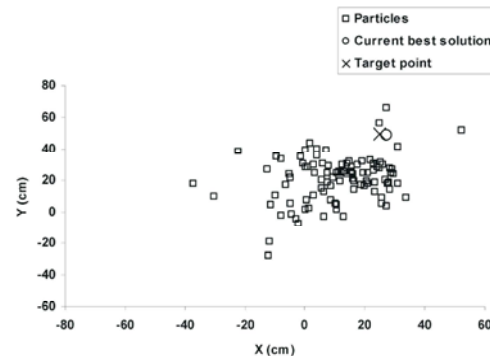


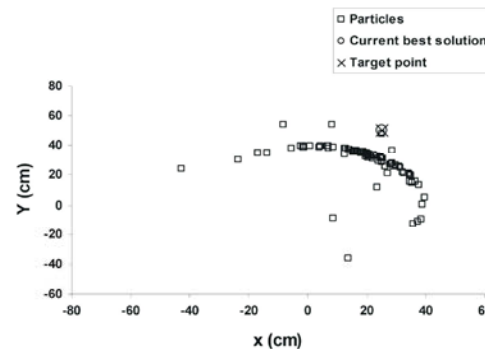
Fig. 3: Convergence errors for EM and GA



(a) Initial population of particles



(b) The particles move towards the target point



(c) Current solution is very close to the target point  
Fig. 4: Convergence the particles by using EM

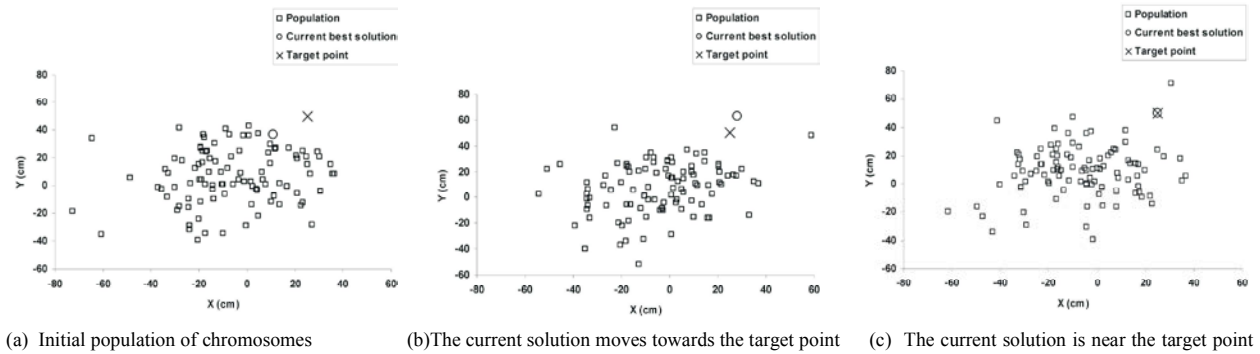


Fig. 5: Convergence the chromosome by using GA

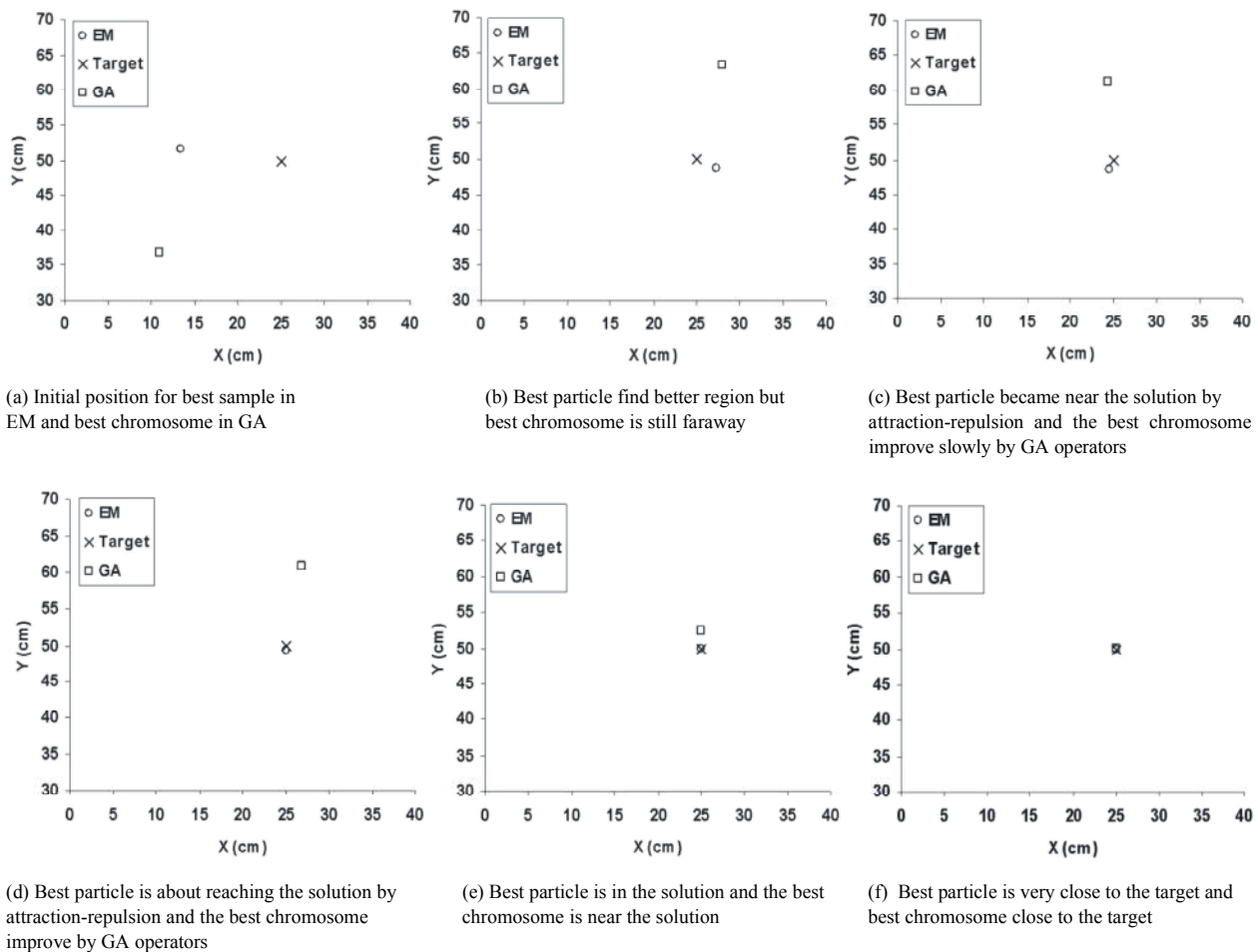
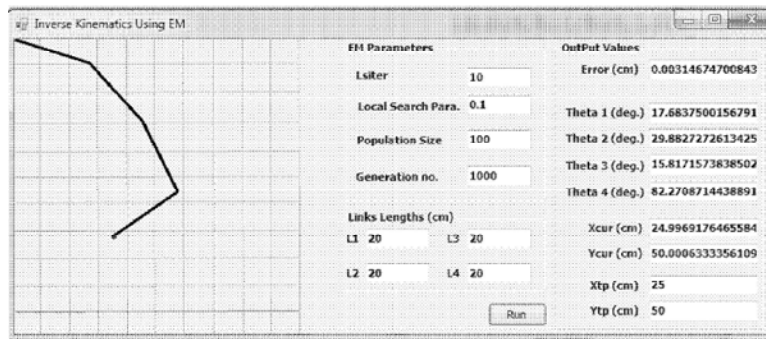


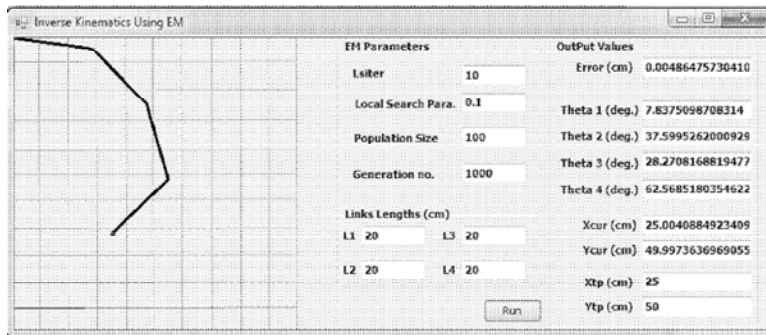
Fig. 6: The convergence of the current best solution to the target point

Fig. 6 (a)-(f) show the convergence of the current best solution by EM algorithm and GA with respect to the target solution. Figures 7 (a)-(c) and Figures 8 (a)-(c) show the multiple solutions in the same point for EM and GA, respectively.

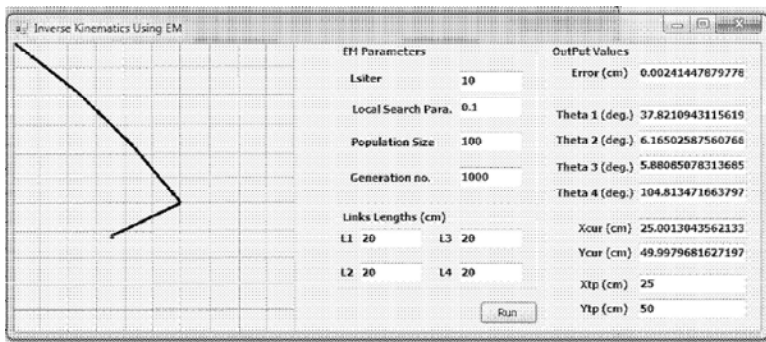
Further results are shown in Table 1 for both algorithms. According to the result of Table 1, the number of iterations for EM algorithm is less than for GA as well as the error produces by GA is bigger than that of EM at each target point.



(a) First configuration by EM

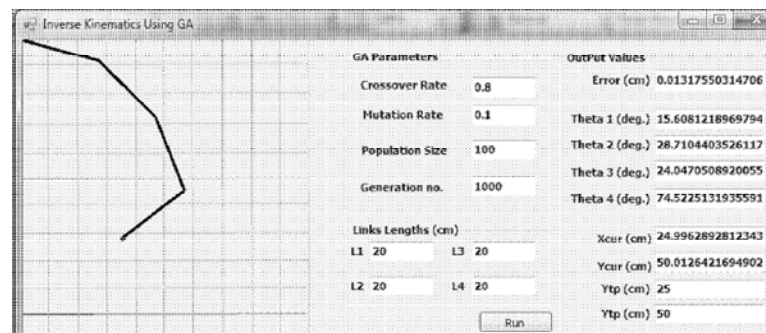


(b) Second configuration by EM

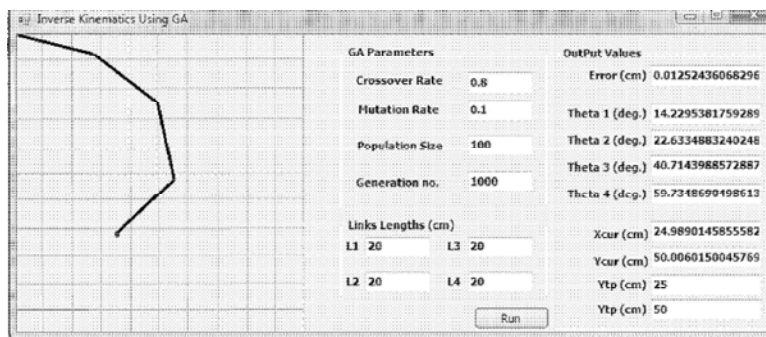


(c) Third configuration by EM

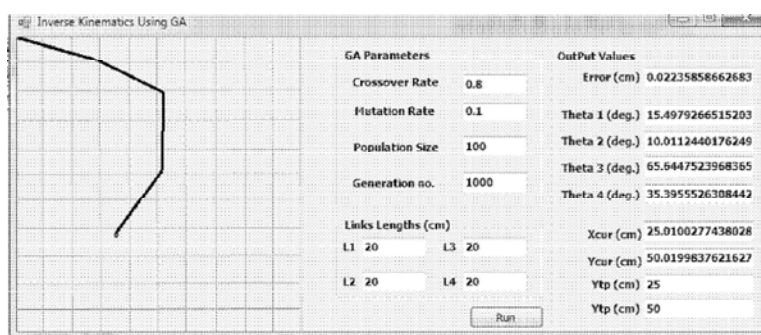
Fig. 7: Some configurations for robot manipulator at the target point by EM



(a) First configuration by GA



(b) Second Configuration by GA



(c) Third Configuration by GA

Fig. 8: Some configurations for robot manipulator at the target point by GA

Table 1 Results for EM and GA

Target position (cm)	Electromagnetism-Like			Genetic Algorithm		
	Current solutions (cm)	Position error (cm)	Generation number	Current solutions (cm)	Position error (cm)	Generation number
(25,50)	(25.000,49.992)	0.0072	400	(24.989,49.991)	0.0129	390
	(25.004,49.991)	0.0094	85	(25.000,50.011)	0.0113	830
	(25.003,49.999)	0.0035	835	(25.009,50.004)	0.0107	910
	(24.995,49.997)	0.0049	185	(24.994,50.010)	0.0116	165
	(24.998,50.000)	0.0019	255	(24.992,50.002)	0.0080	870
(40,30)	(39.997,30.004)	0.0050	125	(39.977,29.966)	0.0398	190
	(40.005,30.003)	0.0064	645	(40.029,30.043)	0.0527	580
	(39.999,29.999)	0.0012	155	(40.013,30.014)	0.0194	335
	(39.998,30.001)	0.0019	215	(40.007,30.014)	0.0162	880
	(40.000,29.995)	0.0041	265	(39.992,29.983)	0.0185	410
(30,40)	(29.995,40.001)	0.0043	40	(29.999,40.019)	0.0194	235
	(29.994,40.003)	0.0065	215	(30.000,39.975)	0.0240	230
	(30.000,39.996)	0.0038	225	(30.001,40.019)	0.0198	650
	(29.999,39.995)	0.0040	650	(30.007,40.000)	0.0078	95
	(39.002,39.999)	0.0021	260	(30.004,40.007)	0.0084	435

## CONCLUSION

IK problem is a very important topic in robotic systems, because it is used for trajectory planning and in the control the motion of the robot. In this paper,

two methods have been tested to check their ability to solve this problem. Both algorithms could find multiple configurations at the desired point. The results show that EM algorithm is more accurate than GA and with less number of iterations than GA. EM algorithm



uses the attraction and repulsion for the particles to converge faster to the feasible region and optimum solution. On the other hand, GA uses its operators to move the individuals directly to the near optimal solution.

## REFERENCES

1. Kalra, P., P.B. Mahapatra and D.K. Aggarwal, 2006. An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. *Mechanism and Machine Theory*, 41: 1213-1229.
2. Chapelle, F. and P. Bidaud, 2004. Closed form solutions for inverse kinematics approximation of general 6R manipulators. *Mechanism and Machine Theory*, 39: 323-338.
3. Tabandeh, S., C. Clark and W. Melek, 2006. A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering. *IEEE Congress on Evolutionary Computation*, pp: 1815-1822.
4. Köker, R., C. öz, T. Çakar and H. Ekiz, 2004. A study of neural network based inverse kinematics solution for a three-joint robot. *Robotics and Autonomous Systems*, 49: 227-234.
5. Chandra, R. and L. Rolland, 2011. On solving the forward kinematics of 3RPR planar parallel manipulator using hybrid metaheuristics. *Applied Mathematics and Computation*, 217: 8997-9008.
6. Debels, D., B.D. Reyck, R. Leus and M. Vanhoucke, 2006. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*, 169: 638-653.
7. Jamili, A., M.A. Shafia and R.T. Moghaddam, 2011. A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem. *Expert Systems with Applications*, 38: 5895-5901.
8. Feng, Y., W.Y. Nan and W.S. Ning, 2011. Inverse kinematic solution for robot manipulator based on electromagnetism-like and modified DFP algorithms. *Acta Automatica Sinica*, 37: 74-82.
9. Yahya, S., M. Moghavvemi and H.A.F. Mohamed, 2011. Geometrical approach of planar hyper-redundant manipulators: Inverse kinematics, path planning and workspace. *Simulation Modelling Practice and Theory*, 19: 406-422.
10. Yao, Z. and K. Gupta, 2007. Path planning with general end-effector constraints. *Robotics and Autonomous Systems*, 55: 316-327.
11. Mitsi, S., K.D. Bouzakis, D. Sagris and G. Mansour, 2008. Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm. *Robotics and Computer- Integrated Manufacturing*, 24: 50-59.
12. Hnaïen, F., X. Delorme and A. Dolgui, 2009. Genetic algorithm for supply planning in two-level assembly systems with random lead times. *Engineering Applications of Artificial Intelligence*, 22: 906-915.
13. Lee, M., 2003. Evolution of behaviors in autonomous robot using artificial neural network and genetic algorithm. *Information Sciences*, 155: 43-60.
14. Köksoy, O. and T. Yalcinoz, 2008. Robust design using Pareto type optimization: A genetic algorithm with arithmetic crossover. *Computers & Industrial Engineering*, 55: 208-218.
15. Li, F., L.D. Xu, C. Jin and H. Wang, 2011. Intelligent bionic genetic algorithm (IB-GA) and its convergence. *Expert Systems with Applications*, 38: 8804-8811.
16. Maciel, S.L. C.O. Castillo and L.T. Aguilar, 2011. Generation of walking periodic motions for a biped robot via genetic algorithms. *Applied Soft Computing*, 11: 5306-5314.
17. Yurtkuran, A. and E. Emel, 2010. A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert Systems with Applications*, 37: 3427-3433.
18. Chang, P.C., S.H. Chen and C.Y. Fan, 2009. A hybrid electromagnetism-like algorithm for single machine scheduling problem. *Expert Systems with Applications*, 36: 1259-1267.
19. Birbil, S.I. and S.C. Fang, 2003. An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25: 263-282.