A New Proposed Pendulum-Like with Attraction-Repulsion mechanism Algorithm for Solving Optimization Problems

I A Abed¹, S P Koh², K S M Sahari², S K Tiong², M M Ali¹ and A A Abdul Kadhim¹
¹Engineering Technical College Basrah, Southern Technical University, Iraq.
²College of Engineering, Universiti Tenaga Nasional, Selangor, Malaysia.

Abstract. This work aims to contribute to the domain of new modification in electromagnetism-like (EM) algorithm, which will be referred to in this work as pendulum-like with attraction-repulsion mechanism algorithm (PA). PA algorithm has been constructed with the purpose of solving inverse kinematics (IK) problem. The proposed new algorithm mimics the concepts of a simple physical pendulum and is derived as a local search in order to promote the performance of the EM algorithm. Comparisons between the suggested PA with the available algorithms, such as genetic algorithm (GA) and EM were conducted using two mathematical test functions. On top of the testing, the PA algorithm is tested to solve the problem of IK for a four degree-of-freedom (DOF) planar robot. The simulation results indicate that the PA algorithm outperforms other approaches in terms of accuracy and the speed of convergence through the errors and the decreasing of objective with iterations as mention in presented tables and figures, on top of being able to successfully solve the IK problem with multiple robot configurations.

Keywords: Simple physical pendulum; electromagnetism-like; attraction-repulsion mechanism; optimization; planar robot.

1. Introduction
Inverse kinematics (IK) problem is a very important topic in robotic systems[1], as it is used for trajectory planning and motion control of the robot. In recent research, artificial intelligent has been applied in order to ease and optimize the solution of IK. Kalra et al. [2] suggested an evolutionary technique for industrial robot in order to solve the inverse kinematics which is depended on genetic algorithm (GA) of single-level. The authors utilized the fitness function in the suggested approach, they explained this function in a way that needs isolation of evaluation of the error of position and the displacement of total joint. Utilizing a niche method, they earned population candidates with arrangement which is satisfied around the multiple solutions of inverse kinematics, this is in a manner of selection binary tournament selection operation is used with restriction mating. The Scara and Puma are two types of robotics used to examine this method. For every point of the target, the first robotic configuration comes up with one or two solution. Four solutions for every location of the wrist are given by the second robotic type. Using niching method 1 and niching method 2, the authors proposed for the wrist a minimum errors of position for the resulted values of the variables of the joint. In niche method 1 for Scara, the range of the errors of position between 0.4-2.19 mm while for niche method 2 within 0.14-0.69 mm. Meanwhile, niche strategy 2 ended up with lesser positional errors compared to niche strategy 1. Moreover, in their work, for Puma with niche method 2 the error is between 0.36-1.91 mm. In order to reach to the optimal solution in the case of Scara robot about 100 generations are
enough. On the other hand about 300 generations are suitable for Puma robot. The solution of inverse kinematics of the industrial manipulators are introduced by Alavandar and Nigam [3] utilizing the Neuro-Fuzzy. As training data in their work, the Cartesian coordinates and angles used to learn the adaptive Neuro-Fuzzy inference system (ANFIS). They utilized the back-propagation (BP) neural network-like structure as performer for fuzzy inference system to execute ANFIS. In order to prove the efficiency of the suggested technique, it is examined by two simulation examples of 2-DOF (degree of freedom) and 3-DOF. The technique of adaptive particle swarm optimization (A-PSO) concerted the equations of kinematics is given by Zang et al. [4]. Inverse kinematics of serial manipulator is solved by this method. Because of the PSO is easily track down in the local optima and needs size with large candidates, the authors changed the classical PSO. The combinatorial matrix deviation from the end of the robot to the goal is the utilized fitness function. The complication of the analysis of inverse kinematics can be decreased by the introduced approach, which renders the adaptive solution obtainable. They compared their method with the traditional PSO algorithm, and the efficiency of the proposed method is demonstrated in the case study. Yin et al. [5] have been developed a method for solving the inverse kinematics problem for robot, especially for those manipulators with high-dimensional nonlinear kinematics equations. They transform the IK problem into a minimization problem. Then, an Electromagnetism-like (EM) method is used to solve this equivalent problem. Moreover, in order to further improve the accuracy of the EM algorithm, a hybrid method, which combines EM with the Davidon-Fletcher-Powell (DFP) method, is proposed. The EM complexity is independent on the characteristics of the kinematics equations, involving dimensionality and the degree of nonlinearity. Moreover, EM can be used as an accompanying algorithm for the DFP method in order to obtain better precision at a lower iteration number. EM algorithm is a good choice in solving continuous problems [5]. It is derived from the attraction-repulsion theory of physics, and is classified as a population based algorithm [6]. It is capable of a faster solution approximation compared to other algorithms [7], and is also fairly accurate [8]. The local search is one of the important steps in EM algorithm. Without it, the particles would lack the local information, which makes it difficult to obtain the local optimums [9]. The efficiency of simple random local search applied is low [10]. It is random, and one of its drawbacks is that it is incapable of evading the local optima [11]. In this paper, a new method contributing to the field of robotics has been proposed to effectively solve the inverse kinematics for robot manipulator. The remainder of the paper is divided as follows. Section 2 and Section 3 provides the definition of EM and the proposed methods; Section 4 details a brief four test functions, which are used for comparison; Simulation results have been presented in section 5 for all methods; and the work will be concluded in Section 6.

2. EM Algorithm

In order to direct the solutions to the optimal position, this algorithm is used. It is utilized the attraction-repulsion technique [12,13,14]. The procedures of EM are set in figure 1 [15,16].

```
1. Initialization
2. iteration ← 1
3. While iteration < Max_iteration do
   4. Evaluation of objective function
   5. Local search (Lsiter, δ)
   6. F ← Calc(F)
   7. Move (F)
   8. iteration ← iteration + 1
9. End while
```

Figure 1. The steps of EM
3. The Proposed Algorithms

3.1. Modified EM Algorithm with Concentrated Local Search

This is the first proposed modification which does not mention before, a population of 10 particles is randomly distributed in the search space with the objective function values shown in figure 2. The current best solution is chosen according to the objective function value at the current moment. Subsequently, the procedure would start with global search by calculating the forces which would be used to move the particles as shown in figure 3. This is because the global search is used to discover the entire search space or its approximation [17]. Then, the Concentration Local Search (CLS) in the neighbourhood of the current best solution is obtained after the global search process. This would help to fine tune the best solution and give better convergence as shown in figure 4. Thereafter, the population would be randomized again while the current best solution is stored for the next generation. The randomization step helps the particle to avoid being stuck in local optima or to obtain a better position solution.

![Figure 2. Initial population](image2)

![Figure 3. Population after the attraction-repulsion mechanism](image3)
Therefore, instead of doing the neighbourhood search for the particles which are far from the target, concentrated search performed for current best particle will be focusing on the solutions in the neighbourhood the best particle. Figure 5 illustrates the flowchart of MEMC algorithm.

The pseudo code for CLS is shown in figure 6. This local search also uses adaptive local search parameter and it will search the neighbourhood for the current best solution only. It will concentrate the search for the current best solution according to the value of counter.
3.2. Pendulum-Like with Attraction-Repulsion Mechanism Algorithm

In order to solve the problems which are complex, a robust method should be provided. The proposed integration between attraction- repulsion mechanism and pendulum-like algorithm is introduced. This also enhance the ability of the searching algorithm with a minimum drawbacks. Hence the procedures of the attraction- repulsion mechanism and movement are still as in the traditional electromagnetism-like. PA algorithm steps are shown in figure 7, while the pseudo code of the method is shown in figure 8.

Figure 6. Pseudo code for Concentrated Local Search algorithm

Figure 7. A block diagram for PA algorithm
Figure 8. The pendulum-like algorithm pseudo code

4. Mathematical Test Functions
In the context of this work, mathematical test functions are utilized for the purpose comparing the performance of proposed algorithms. But there is no standard list of test functions one has to follow [18]. For the subsequent subsections, the details of each of the test functions utilized would be discussed in brief [18], where a two-dimensional plot of two common functions which are:

4.1. De Jong’s Function
De Jong’s function is defined as in equation 1, while the two-dimensional plot is shown in figure 9.

\[ f_1(x) = \sum_{i=1}^{n} x_i^2 \]  

(1)

And \( f_1(x^{\text{opt}}) = 0 \) at \( x^{\text{opt}} = (0,0,\ldots,0) \). While \(-5.12 \leq x_i \leq 5.12, \ i = 1,\ldots,n\).
Figure 9. A two-dimensional plot of De Jong’s function

4.2. Rastrigin’s Function
Rastrigin’s function is mathematically defined as in equation 2, while the two-dimensional plot is shown in figure 10.

\[ f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10) \]  

(2)

And \( f_2(x^{opt}) = 0 \) at \( x^{opt} = (0,0,\ldots,0) \). While \(-5.12 \leq x_i \leq 5.12, i = 1,\ldots,n\).

![Figure 10. The function of Rastrigin](image)

4.3. Axis parallel hyper-ellipsoid’s function
The axis parallel hyper-ellipsoid function with

\[ f_3(x) = \sum_{i=1}^{n} (i.x_i^2) \]  

(3)

And \( f_3(x^{opt}) = 0 \) at \( x^{opt} = (0,0,\ldots,0) \). While \(-5.12 \leq x_i \leq 5.12, i = 1,\ldots,n\), the two-dimensional plot for the function is shown in figure 11.

![Figure 11. The function of axis parallel hyper-ellipsoid function](image)
4.4. Rosenbrock’s function

The Rosenbrock’s function with

\[ f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \]  

And \( f_4(x^{opt}) = 0 \) at \( x^{opt} = (1,1,\ldots,1) \). While \(-2.048 \leq x_i \leq 2.048, i = 1,\ldots,n\), the two-dimensional plot for the function is shown in figure 12.

![Figure 12. The function of Rosenbrock](image)

5. Simulation Results

The development of EM algorithm and pendulum-like with attraction-repulsion mechanism algorithm, has been performed in the previous sections. In this section, the simulation results for all methods exactly the PA algorithm has been presented and compared. The details required are shown in table 1, where the specifications of GA are selected after many tests.

| Software               | Visual Basic. net |
|------------------------|-------------------|
| PC                     | Intel Core 2 Due 2.1GHz 2.1 GHz |
| Population size for all methods | 50 GA |
| Crossover              | Arithmetic |
| Mutation               | Uniform          |
| Selection              | Roulette wheel   |
| Crossover rate         | 0.8              |
| Mutation rate          | 0.04             |

5.1. Rosenbrock’s function

The number of iterations for De Jong’s and axis parallel hyper-ellipsoid function test is 100 while for Rastrigin’s and Rosenbrock’s function test is 1000 for all techniques. The local search iterations are set at 100 for EM, MEMA, and MEMC, while the local search parameter for EM is 0.001 in the case of De Jong’s, axis parallel hyper-ellipsoid, and Rosenbrock’s function and 0.01 in the case of Rastrigin’s function. They are selected after many runs. The parameter is set to 2 for De Jong’s function test, 1 for Rastrigin’s function test, 2.5 for axis parallel hyper-ellipsoid function test, and 1.1 for Rosenbrock’s function test for MEMA, MEMC, and PA algorithms. In PA algorithm the gravity constant is also 1 for all tests while the particle mass is 0.05 in the case of De Jong’s function, 1.2 in
the case of Rastrigin’s function, 0.1 in the case of axis parallel hyper-ellipsoid function, and 0.027 in the case of Rosenbrock’s function.

Figure 13. Algorithms evaluation on De Jong’s function

The results for test functions simulations are shown in figures 13-16 and table 2. In De Jong’s function, from start until about 30 iterations, GA and EM algorithm are very close to each other. After that, EM algorithm improved better than GA which continued to improve very slowly. On the other hand, MEMC algorithm was faster and slightly better than MEMA algorithm. The superiority of PA algorithm over all the other methods shows the advantage of using Pendulum-like algorithm to enhance the performance, where the average value of objective function is 3.02E-17.

Figure 14. Algorithms evaluation on Rastrigin’s function
For the Rastrigin’s function in figure 14, EM algorithm and GA are near to each other for up to about 600 iterations. Then, EM algorithm converges very rapidly at about 630 iterations until it reaches $6.57 \times 10^{-7}$. Although MEMC converges faster than MEMA at very early iterations, MEMA achieved better accuracy of $1.71 \times 10^{-8}$ after 1000 iterations. Overall, PA algorithm performed the best among all the methods with the objective value of $2.62 \times 10^{-10}$.

Figure 15. Algorithms evaluation on axis parallel hyper-ellipsoid function

For the axis parallel hyper-ellipsoid function in figure 15, up to about 65 iterations, EM and GA are comparable. MEMC converges faster than MEMA and achieved slightly better accuracy of $2.72 \times 10^{-14}$. PA algorithm outperforms all the other methods and even if at very early iterations it seems to be comparable to MEMC but it is much less processing time.

Figure 16. Algorithms evaluation on Rosenbrock’s function
In figure 16, EM and GA are comparable up to about 200 iterations. MEMC is much better than MEMA, EM, and GA in this test most probably because of the Concentrated Local Search operator which is used to enhance the solution further. PA algorithm performs the best between all the other methods which shows the powerful of using the Pendulum-like algorithm.

Table 2. The average value and the Standard Deviation for each of the algorithm based on the given test function

| Algorithm | De Jong’s function | Rastrigin’s function | Axis parallel hyper-ellipsoid function | Rosenbrock’s function |
|-----------|--------------------|----------------------|----------------------------------------|-----------------------|
|           | Average value      | SD                   | Average value                          | SD                    |
| GA        | 5.58E-06           | 4.24E-06             | 6.21E-03                               | 2.56E-06              |
| EM        | 4.13E-11           | 2.37E-11             | 2.53E-07                               | 3.96E-11              |
| MEMC      | 1.07E-12           | 1.33E-14             | 1.57E-11                               | 4.52E-16              |
| Proposed PA | 3.02E-17         | 2.16E-17             | 2.62E-10                               | 3.40E-18              |

5.2. Simulation Results of Inverse Kinematics

The number of iterations in the proposed methods is 1000. The parameter $\delta$ is 0.00001 while the chosen value of $Lsiter$ is 100 after several runs. The parameters $\beta, \mu_p$, and $g$ are 2, 0.05, and 1, respectively; which were also obtained after running some tests. Figures 17 and 18 and table 3 show the comparison between GA, EM, MEMA, MEMC, and PA for two suggested goal points. Figure 17 were captured after solving the IK using the above mentioned methods at the end effector position which is (100, 50) cm where this point is selected according to the robot workspace and suitable with the robot dimensions, GA converges faster than EM algorithm for about the first 400 iterations. Then, the EM algorithm enhances rapidly from about 500 to 600 iterations. After that, it converges slowly and reaches average error of 1.69E-6 cm. MEMA and MEMC algorithms are close to each other with better convergence for MEMC at the beginning of the simulation and slightly better accuracy for MEMA algorithm at the end of the simulation. PA algorithm is outperformed the others in terms of high speed convergence and very low positional error which is 3.50E-9 cm.

![Figure 17. Algorithms evaluation at position (100, 50) cm](image)
The performances of the GA and EM, MEMA, MEMC, and PA algorithms at the task point of (50, 60) cm which is a new point and suitable for the situation of the robot are shown in figure 18 and table 3. The worse result is obtained by GA which is 4.13E-3 cm while best algorithm with an accuracy of 1.37E-9 cm is PA algorithm.

![Figure 18. Algorithms evaluation at position (50, 60) cm](image)

**Table 3.** Average value and Standard Deviation for each suggested algorithm to solve IK problem

| Algorithm | Target position (100,50) cm | Target position (50,60) cm |
|-----------|-----------------------------|----------------------------|
|           | Average value | SD | Average value | SD |
| GA        | 3.97E-3        | 2.56E-3 | 4.13E-3        | 2.77E-3 |
| EM        | 1.69E-6        | 1.07E-6 | 7.75E-7        | 5.29E-7 |
| MEMC      | 2.11E-7        | 1.61E-7 | 6.21E-8        | 3.20E-8 |
| Proposed PA | 3.50E-9    | 3.44E-9 | 1.37E-9        | 1.31E-9 |

The proposed techniques have been further compared using other Cartesian coordinates within the robot workspace with positional error shown in table 4.

**Table 4.** Results for GA, EM, MEMC, and PA

| Target position (cm) | GA | EM | MEMC | PA |
|----------------------|----|----|------|----|
| (70,20)              | 2.02E-3 | 1.16E-07 | 3.37E-08 | 2.78E-10 |
| 2.22E-3              | 2.05E-07 | 2.89E-08 | 4.29E-10 | 2.43E-10 |
| 2.87E-3              | 1.39E-07 | 3.17E-08 | 3.27E-10 | 2.32E-10 |
| 4.34E-3              | 2.12E-07 | 3.11E-08 | 2.32E-10 | 2.32E-10 |
| (80,30)              | 2.97E-3 | 1.04E-07 | 3.24E-08 | 5.14E-10 |
| 2.22E-3              | 1.86E-07 | 3.29E-08 | 5.48E-10 | 5.48E-10 |
| 2.98E-3              | 1.45E-07 | 3.77E-08 | 3.92E-10 | 3.92E-10 |
| 1.40E-3              | 2.03E-07 | 3.43E-08 | 5.30E-10 | 5.30E-10 |
| 1.83E-3              | 2.09E-07 | 3.91E-08 | 5.50E-10 | 5.50E-10 |
| (90,30)              | 2.94E-3 | 2.40E-07 | 4.91E-08 | 5.31E-10 |
| 4.37E-3              | 3.31E-07 | 4.08E-08 | 4.06E-10 | 4.06E-10 |
| 7.67E-3              | 2.86E-07 | 4.34E-08 | 5.08E-10 | 5.08E-10 |
| 2.65E-3              | 2.82E-07 | 3.32E-08 | 4.73E-10 | 4.73E-10 |
5.3. Effect of PA on the Behavior of the Population during the Optimization

The number of iterations of PA algorithm for this test is 1000. The constant $\alpha$ is set to be 2. The point mass and the gravity constant are 0.05 and 1, respectively. The first test was carried out on the target point of (100, 50) cm. Figures 19 (a), 19 (b), and 19 (c) show the whole population movement towards optimal position. Figures 20 (a), 20 (b), and 20 (c) show the behavior of the algorithm at the second target, which is (90, 60) cm. At beginning, the workspace will initialize with new random population. Search the convergence of the best particle will happen after a few generations, when the algorithm get about back and forth. The desired position will be near the current best solution and a group of particles. So as to get so near solution to the desired point in every generation, the produced solution which has the superiority to the current best solution, will be changed with the current best solution.

(a) Distribution of initial particles for the first end effector position

(b) Particles movement towards the target for the first end effector position
(c) The goal is very near to the current best solution, and some other particles observe better region for the first end effector position.

**Figure 19.** Behavior of particles using PA algorithm for the first end effector position.

(a) Initial particles population

(b) New particles positions are generated
**Conclusion**

EM algorithm is an effective optimization method. However, it requires better performance local search in order to fine tune and further improve its result. Instead of using additional gradient or other local search, a powerful and newly developed local search namely pendulum-like algorithm has been replaced the local search of original EM and produced the PA algorithm. This algorithm has been tested on mathematical test functions and also used to solve for IK in planar robot. In both cases, the results of PA algorithm are compared with original EM algorithm and GA. The proposed PA algorithm performs better, and it is not easily trapped in local optima, due to the combination of directed force with the back and forth principle. In this research, a simulator package has been designed and developed in order to illustrate the continuous enhancement in PA solutions by some iterations, in the inverse kinematics. As a conclusion, the proposed PA is able to provide different solutions at each task point of the robot with higher accuracy and faster convergence speed. Hence, different applications have been given here for proposed PA which are the mathematical functions and IK. However, additional applications can be solved by this algorithm such as the optimal path for robot , optimal parameters for the photovoltaic model.

**References**

[1] Abed, I. A., Koh, S. P., Sahari, K. S. M., Tiong, S. K., & Yap, D. F. 2012 Comparison between genetic algorithm and electromagnetism-like algorithm for solving inverse kinematics. *World Applied Sciences Journal*, 20(7), 946-954.

[2] Kalra, P., Mahapatra, P. B., & Aggarwal, D. K. 2006 An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. *Mechanism and Machine Theory*, 41, 1213-29.

[3] Alavandar, S., & Nigam, M. J. 2008 Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators. *Int. J. of Computers, Communications & Control*, 3, 224-234.

[4] Zhang, P., Mu, X., Ma, Z., & Du, F. 2012 An adaptive PSO-based method for inverse kinematics analysis of serial manipulator. In *International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, IEEE, pp. 1122-26.

[5] Yin, F., Wang, Y. N., & Wei, S. N. 2011 Inverse kinematic solution for robot manipulator based on electromagnetism-like and modified DFP algorithms. *Acta Automatica Sinica*, 37, 74-82.
[6] Khalili, M., & Tavakkoli-Moghaddam, R. (2012). A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem. *Journal of Manufacturing Systems, 31*, 232-239.

[7] Lee, C. H., Chang, F. K., & Lee, Y. C. 2010 An improved electromagnetism like algorithm for recurrent neural fuzzy controller design. *International Journal of Fuzzy Systems, 12*, 280-290.

[8] Birbil, Ş. İ., Fang, S. –C. 2003 An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization, 25*, 263-282.

[9] Guan, X., Dai, X., Qiu, B., & Li, J. 2012 A revised electromagnetism-like mechanism for layout design of reconfigurable manufacturing system. *Computers & Industrial Engineering, 63*(1), 98-108.

[10] Zhang, K., & Cao, P. 2011 Modified electromagnetism-like algorithm and its application to slope stability analysis. *Journal of Central South University of Technology, 18*, 2100-2107.

[11] Lin, J. L., Wu, C. H., & Chung, H. Y. 2012 Performance Comparison of Electromagnetism-Like Algorithms for Global Optimization. *Applied Mathematics, 3*, 1265-1275.

[12] Su, C.T. and H.C. Lin, 2011. Applying electromagnetism-like mechanism for feature selection. *Information Sciences, 181*(5): 972-986.

[13] Filipović, V., A. Kartelj and D. Matić, 2013. An electromagnetism metaheuristic for solving the Maximum Betweenness Problem. *Applied Soft Computing, 13*: 1303-1313.

[14] Abed, I. A., Sahari, K. S. M., Koh, S. P., Tiong, S. K., & Jagadeesh, P. 2013 Using Electromagnetism-like algorithm and genetic algorithm to optimize time of task scheduling for dual manipulators. In *2013 IEEE Region 10 Humanitarian Technology Conference* (pp. 182-187). IEEE.

[15] Jolai, F., R. Tavakkoli-Moghaddam, A. Golmohammadi and B. Javadi, 2012. An Electromagnetism-like algorithm for cell formation and layout problem. Expert Systems with Applications, *39*(2): 2172-82.

[16] Abed, I. A., Koh, S. P., Sahari, K. S. M., Tiong, S. K., & Tan, N. M. 2014 Solving the inverse kinematics for robot manipulators using modified electromagnetism-like algorithm with record to record travel. *Research Journal of Applied Sciences, Engineering and Technology, 7*(19), 3986-94.

[17] Talbi, E. 2009 *Metaheuristics: from design to implementation*. Hoboken, New Jersey: John Wiley & Sons.

[18] Yang, X. -S. 2010 *Test problems in optimization*. In Xin-She Yang (Eds.), Engineering Optimization: An Introduction with Metaheuristic Applications. Hoboken, New Jersey: John Wiley & Sons.