

Inverse Kinematics of SCARA Manipulator based on SFLA

Duc Hoang Nguyen
Faculty of Electrical and Electronics Engineering
HCMC University of Technology
Ho Chi Minh City, Vietnam

ABSTRACT

In this paper, SCARA manipulator was designed for the cutting operation. Due to cutting path is nearly closed, normal SCARA manipulators cannot track this path without colliding with the cutting tool. In this study, the author suggested placing SCARA manipulator on a moving axis. This leads to finding inverse kinematics of moving SCARA manipulator is very complicated. So, the author used Shuffled Frog Leaping Algorithm (SFLA) to obtain the solutions of the inverse kinematics equation. The SFLA is a bio-inspired optimization method that consists of a frog leaping rule for local search and a memetic shuffling rule for global information exchange. The simulation results showed that the SFLA could be effectively used for obtaining the inverse kinematics solutions of the manipulator.

General Terms

Algorithms

Keywords

Optimization, SFLA, Inverse kinematics, SCARA manipulator

1. INTRODUCTION

The inverse kinematics is one of the most important problems in robotics, especially in trajectory planning of robots. There are mainly three techniques for this problem, namely, geometric methods, algebraic methods, and iterative methods [1]. These methods are only used in some cases, such as when the first three joints of the manipulator have a closed-form solution. When geometry of the structure become complex or having obstacles, most of these methods are unable to determine the inverse kinematics solution of the robotic manipulator efficiently. Hence, intelligent algorithms are developed in order to solve the inverse kinematics solutions of robotic manipulators.

A number of intelligent methods have been proposed to solve the inverse kinematics solutions of the manipulator. In [2], a Genetic Algorithm (GA) for solving the inverse kinematics of a serial robotic manipulator is presented. The algorithm is capable of finding multiple solutions of the inverse kinematics through niching methods. In [3], authors presented a particle swarm optimization algorithm (PSO) for solving the inverse kinematics of 7 degree-of-freedom (DOF) robotic manipulators. Simulation results are conducted to illustrate the effectiveness and merit of the proposed methods. In the article [4], authors introduced an approach for solving the inverse kinematics of mobile manipulators based on differential evolution (DE). Simulation and real experiments were performed to prove the effectiveness of the proposed approach.

In this paper, the author introduces an application of the Shuffled Frog Leaping Algorithm for solving the inverse kinematics solutions of the manipulator. The focus is to find the inverse kinematics solution of SCARA manipulator such that don't collide with cutting tool. SFLA is a swarm-based optimization method and has been successfully applied to solve various optimization problems [5-10].

The remaining paper is organized as follows: Section 2 introduces the SCARA manipulator and its kinematics. Section 3 presents overview of the Shuffled Frog Leaping Algorithm. The description how to find the inverse kinematics solutions of the manipulator from cutting path is given in section 4. Section 5 shows the obtained results and Section 6 concludes this paper.

2. KINEMATICS OF SCARA MANIPULATOR

The model of the SCARA manipulator is used in this paper as Fig. 1. It's a 5-DOF robot manipulator.

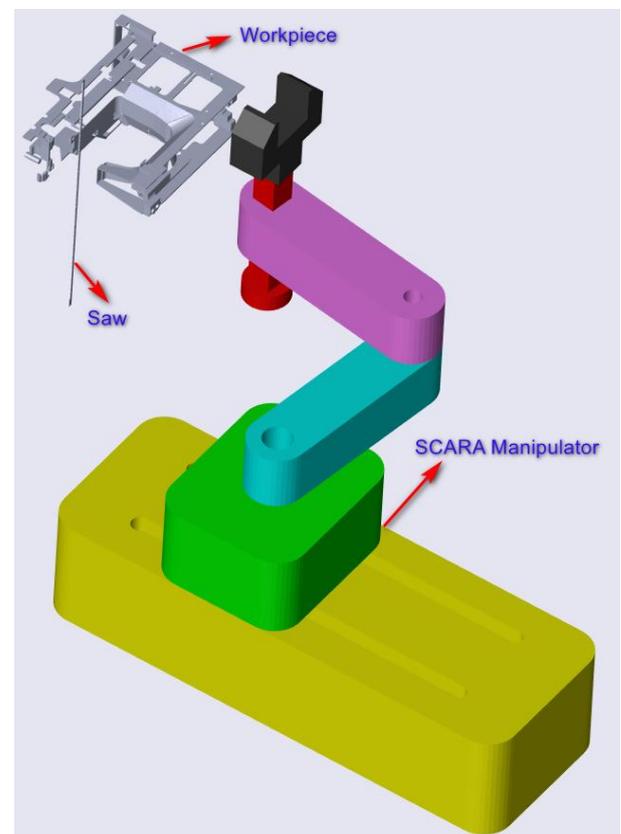


Fig. 1: Prototype of a SCARA manipulator

2.1 Forward kinematics

SCARA manipulator in this paper has 5 degrees of freedom (DOF). It's composed of 4 links connected by 5 joints. The forward kinematics of SCARA manipulator is analyzed by the Denavit–Hartenberg (DH) method, which finds the end-effector position based on the coordinates of the joints.

Table 1 shows the Denavit–Hartenberg (DH) parameters for the SCARA manipulator using the coordinate system as Fig. 2.

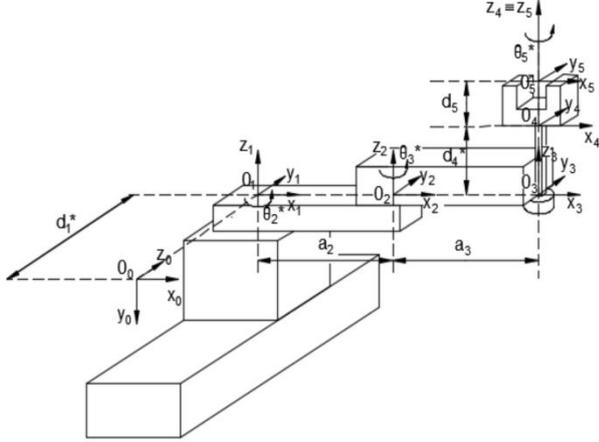


Fig. 2: Coordinate systems attached at each joint of SCARA manipulator

Table 1. DH parameters of 5-DOF SCARA manipulator

i	θ_i	α_i	a_i	d_i
1	0	90^0	0	d_1^*
2	θ_2^*	0	a_2	0
3	θ_3^*	0	a_3	0
4	0	0	0	d_4^*
5	θ_5^*	0	0	d_5

Transformation matrices between the adjacent joint coordinates for the 5-DOF SCARA manipulator are given as:

$${}^4T_5 = \begin{bmatrix} c\theta_5 & 0 & s\theta_5 & 0 \\ s\theta_5 & 0 & -c\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^3T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} * {}^4T_5 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_4 + d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^2T_5 = \begin{bmatrix} c\theta_3 & s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & -c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * {}^3T_5 = \quad (3)$$

$$\begin{bmatrix} c(\theta_3 + \theta_5) & -s(\theta_3 + \theta_5) & 0 & a_3c\theta_3 \\ s(\theta_3 + \theta_5) & c(\theta_3 + \theta_5) & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & d_4 + d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_5 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * {}^2T_5 = \quad (4)$$

$$\begin{bmatrix} c(\theta_2 + \theta_3 + \theta_5) & -s(\theta_2 + \theta_3 + \theta_5) & 0 & a_2c\theta_2 + a_3c(\theta_2 + \theta_3) \\ s(\theta_2 + \theta_3 + \theta_5) & c(\theta_2 + \theta_3 + \theta_5) & 0 & a_2s\theta_2 + a_3s(\theta_2 + \theta_3) \\ 0 & 0 & 1 & d_4 + d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * {}^1T_5 =$$

$$\begin{bmatrix} c(\theta_2 + \theta_3 + \theta_5) & -s(\theta_2 + \theta_3 + \theta_5) & 0 & a_2c\theta_2 + a_3c(\theta_2 + \theta_3) \\ 0 & 0 & -1 & -d_4 - d_5 \\ s(\theta_2 + \theta_3 + \theta_5) & c(\theta_2 + \theta_3 + \theta_5) & 0 & d_1 + a_2s\theta_2 + a_3s(\theta_2 + \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The position and orientation of the end-effector in terms of the joint variables ($\theta_2, \theta_3, \theta_5, d_1, d_4$) can be determined by using following Eq. 6.

$${}^0T_5 = \begin{bmatrix} \eta_x & o_x & a_x & p_x \\ \eta_y & o_y & a_y & p_y \\ \eta_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Each parameter in the matrix of equation Eq. 6 is calculated as follows:

$$\begin{aligned} \eta_x &= c(\theta_2 + \theta_3 + \theta_5) \\ \eta_y &= 0 \\ \eta_z &= s(\theta_2 + \theta_3 + \theta_5) \\ o_x &= -s(\theta_2 + \theta_3 + \theta_5) \\ o_y &= 0 \\ o_z &= c(\theta_2 + \theta_3 + \theta_5) \\ a_x &= 0 \\ a_y &= -1 \\ a_z &= 0 \\ p_x &= a_2c\theta_2 + a_3c(\theta_2 + \theta_3) \\ p_y &= -d_4 - d_5 \\ p_z &= d_1 + a_2s\theta_2 + a_3s(\theta_2 + \theta_3) \end{aligned} \quad (7)$$

2.2 Inverse kinematics

In this paper, the position and orientation of the end-effector are calculated from cutting path and the joint variables are determined as flow chart Fig. 3. From cutting path, cutting points and orientation will be calculated and transformed to the end-effector coordinates. From these values, the inverse kinematics solutions ($\theta_2, \theta_3, \theta_5, d_1, d_4$) of the SCARA manipulator will be determined.

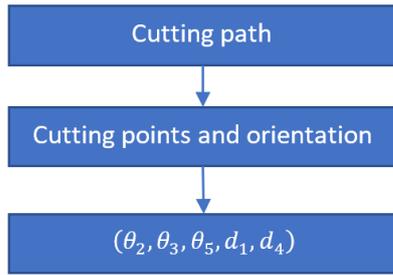


Fig. 3: Calculating joint variables

There are many methods used to determine these solutions. Typically, they are geometric, algebraic, and iterative algorithms. In this study, SFLA is used to solve inverse kinematics problems of SCARA manipulators. The details are given in section 4.

3. SHUFFLED FROG LEAPING ALGORITHM (SFLA)

The SFLA is a bio-inspired optimization method that mimics the memetic evolution of a group of frogs when seeking for the location that has the maximum amount of available food as illustrated in Fig 4.

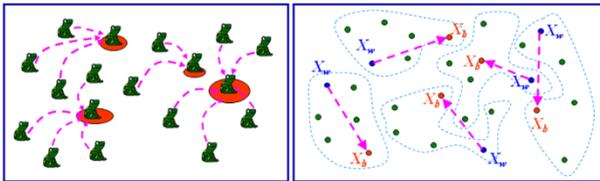


Fig. 4: Evolutionary process of frogs

The algorithm contains the elements of the local search and global information exchange. The SFLA involves a population of the possible solutions defined by a set of virtual frogs that is partitioned into subsets referred to as memeplexes. Within each memeplex, the frogs with the best and the worst fitness are identified as X_b and X_w , respectively. Also, the frog with the global best fitness is identified as X_g . During memeplex evolution, the worst frog X_w leaps toward the best frog X_b . The SFLA performs simultaneously an independent local search in each memeplex using a particle swarm optimization-like method. To ensure global exploration, after a defined number of memeplex evolution steps (i.e. local search iterations), the virtual frogs are shuffled and reorganized into new memeplexes in a technique similar to that used in the shuffled complex evolution algorithm.

In addition, to provide the opportunity for random generation of the improved information, random virtual frogs are generated and substituted in the population if the local search cannot find better solutions. The local searches and the shuffling processes continue until defined convergence criteria are satisfied. The flowchart of the SFLA is illustrated in Fig 5.

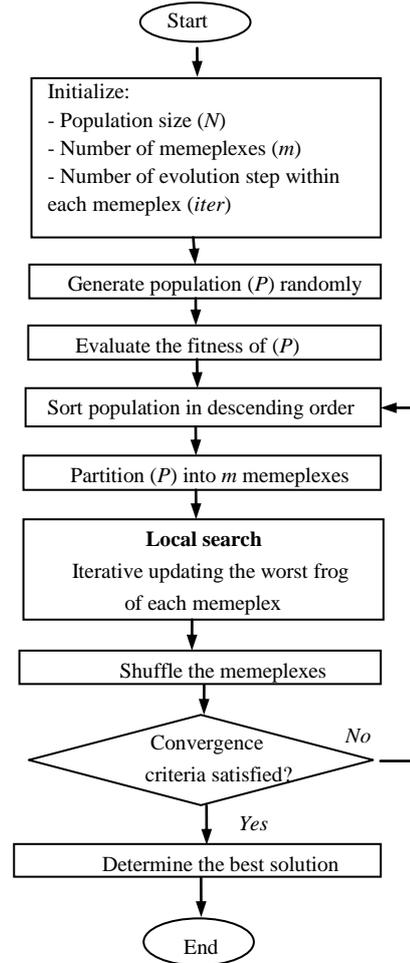


Fig. 5. Flowchart of the SFLA

The idea updating frog leaping rule which is expressed as:

$$D = r \cdot c(X_b - X_w) \quad (8)$$

$$X_w(new) = X_w + D, \|D\| \leq D_{max} \quad (9)$$

Where r is a random number between 0 and 1; c is a constant chosen in the range between 1 and 2.

4. TRAJECTORY DEFINED BY POINTS

This section discusses the procedures for finding the inverse kinematics solutions of the SCARA manipulator.

4.1 Cutting trajectory

In order to control SCARA manipulator to move workpiece along to fixed cutting tool, we need to transform the coordinates of cutting points into coordinate of the end-effector. Besides, moving direction must be tangential to cutting tool.

In the Fig. 6, the cutting path is red path while the cutting tool is green point. With cutting path as Fig. 6, after coordinate transformation, we have end-effector trajectory as Fig. 7.

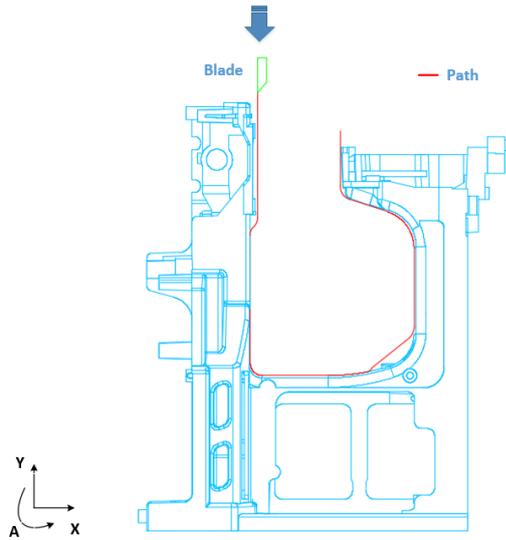


Fig. 6. Cutting path on the workpiece

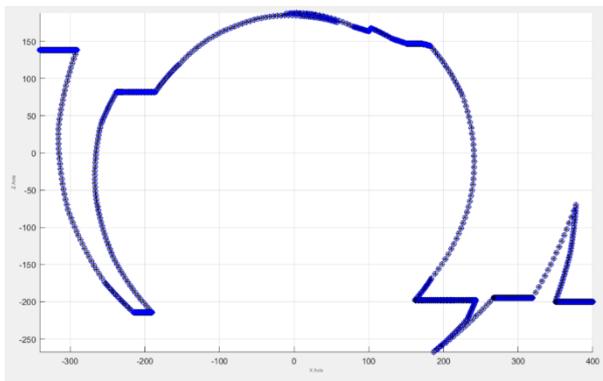


Fig. 7. End-effector trajectory

4.2 Computation of the inverse kinematics solution using SFLA

Computing process of the inverse kinematics solution of SCARA manipulator using SFLA is shown in Fig. 8.

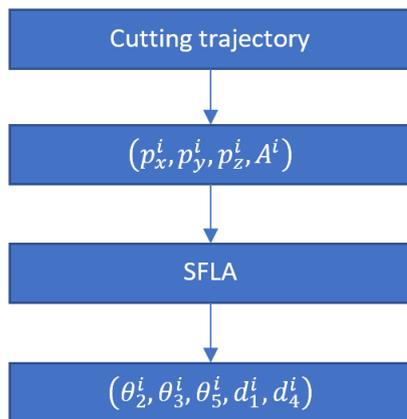


Fig. 8: Computing the inverse kinematics using SFLA

This process as follows: dividing the cutting trajectory into N points and transforming the coordinate of these points into the pose of the end-effector $(p_x^i, p_y^i, p_z^i, A^i)$. Then, SFLA is used to find the angles and position of the joints $(\theta_2^i, \theta_3^i, \theta_5^i, d_1^i, d_4^i)$ such that minimizing the objective function Eq. 10.

$$f = \sqrt{(p_x^* - p_x)^2 + (p_y^* - p_y)^2 + (p_z^* - p_z)^2 + (A^* - A)^2} \quad (10)$$

Where (p_x, p_y, p_z, A) are calculated according to the forward kinematics (7) with notice $A = \theta_2 + \theta_3 + \theta_5$, $(p_x^*, p_y^*, p_z^*, A^*)$ represents the coordinate and orientation values of cutting path.

5. SIMULATION RESULTS

5.1 Simulation parameters

The DH parameters of the SCARA manipulator are given as follows:

$$a_2 = 300\text{mm}, a_3 = 300\text{mm}, d_5 = 80\text{mm}$$

The parameter settings of SFLA are shown in Table 2.

Table 2. The SFLA parameter settings

N	G	C	m	iter	D_{\max}
20	20	2	10	10	∞

5.2 Results and discussion

Fig. 9. is the end-effector trajectory when SCARA manipulator moves workpiece along to cutting path. Fig. 10 and Fig. 11 are the errors of the X, Y coordinates. It can be seen that these errors are very small, which illustrates the accuracy of this algorithm. That shows that the SFLA can be used to solve the inverse kinematics solution of SCARA manipulator.

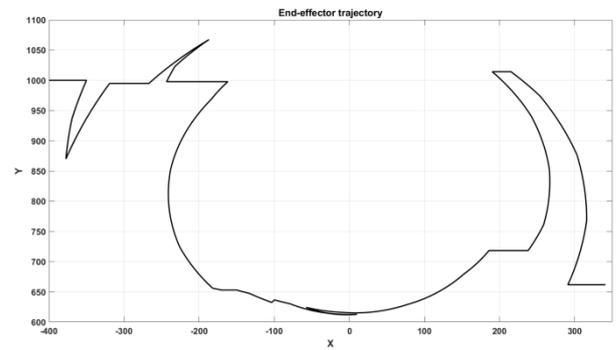


Fig. 9: End-effector trajectory

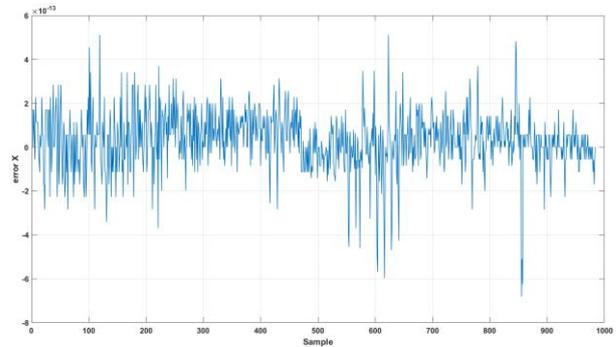


Fig. 10: Position errors of the X coordinate

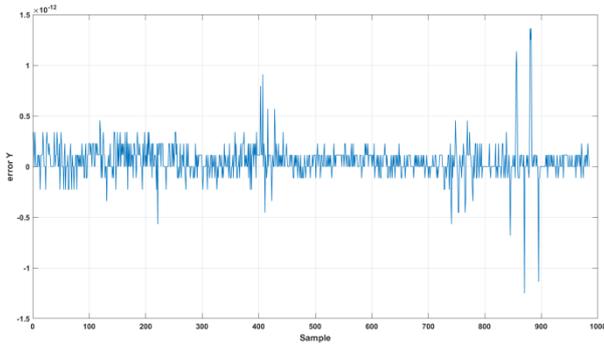


Fig. 11: Position errors of the Y coordinate

Moving trajectory of the joints as Fig. 12 and Fig. 13. Fig. 14. and Fig. 15 present some postures of the SCARA manipulator when moving workpiece along to cutting path. By moving SCARA manipulator on one axis, the links of the manipulator don't collide with the cutting tool.

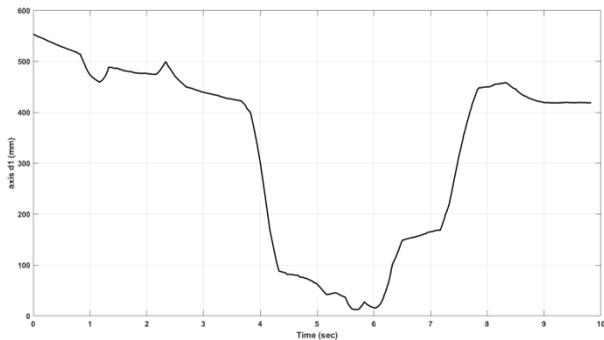


Fig. 12: Position of joint 1

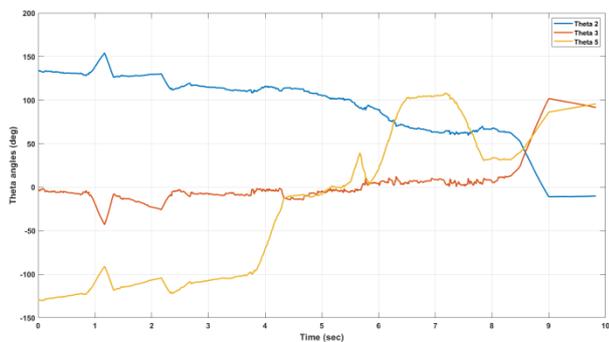


Fig. 13: Angles of joint 2, 3, 5

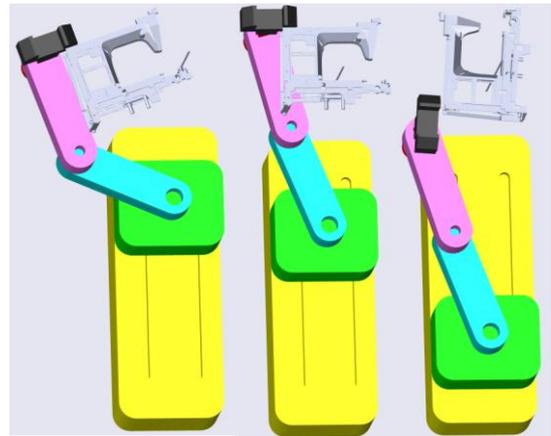


Fig. 14: Some postures of SCARA manipulator



Fig. 15: Some postures of SCARA manipulator (cont.)

6. CONCLUSION

In this paper, the SFLA is proposed to obtain the inverse kinematics solutions of the 5-DOF SCARA manipulator for cutting application. The simulation results show that SFLA gives exact joint angles to control manipulator tracking the cutting path.

Future work will use SFLA for trajectory planning to control the SCARA manipulator that smoothly tracks the cutting path. In addition, a modified SFLA is also proposed for the inverse kinematics problem of robot manipulators.

7. ACKNOWLEDGMENT

We acknowledge the support of time and facilities from HoChi Minh University of Technology (HCMUT), VNU-HCM for this study.

8. REFERENCES

- [1] Ahmed El-Sherbiny, Mostafa A. Elhosseini, Amira Y. Haikal. (2017). A comparative study of soft computing methods to solve inverse kinematics problem. *Ain Shams Engineering Journal*.
- [2] Saleh Tabandeh, Christopher Clark, and William Melek. (2006). A Genetic Algorithm Approach to solve for Multiple Solutions of Inverse Kinematics using Adaptive Niching and Clustering. *IEEE Congress on Evolutionary Computation*.
- [3] Hsu-Chih Huang; Chien-Po Chen; Pei-Ru Wang. (2012). Particle swarm optimization for solving the inverse kinematics of 7-DOF robotic manipulators. *IEEE*

International Conference on Systems, Man, and Cybernetics (SMC).

- [4] Carlos López-Franco, et al. (2017). Inverse kinematics of mobile manipulators based on differential evolution. *International Journal of Advanced Robotic Systems*.
- [5] Muzaffar Eusuff, Kevin Lansey and Fayzul Pasha. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization Vol. 38, No. 2*, 129–154.
- [6] A. Darvishi, A. Alimardani, B. Vahidi, S.H. Hosseinian. (2014). Shuffled Frog-Leaping Algorithm for Control of Selective and Total Harmonic Distortion. *Journal of Applied Research and Technology, Volume 12, Issue 1*, 111–121.
- [7] Kaushik Kumar Bhattacharjee, S.P. Sarmah. (2014). Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. *Applied Soft Computing, Volume 19*, 252–263.
- [8] Yi Han, et al. (2013). Shuffled Frog Leaping Algorithm for Preemptive Project Scheduling Problems with Resource Vacations Based on Patterson Set. *Journal of Applied Mathematics*.
- [9] Daniel Mora-Melia, Pedro L. Iglesias-Rey, F. Javier Martínez-Solano and Pedro Muñoz-Velasco. (2016). The Efficiency of Setting Parameters in a Modified Shuffled Frog Leaping Algorithm Applied to Optimizing Water Distribution Networks. *Water 2016, 8, 182, MDPI*.
- [10] Dina M. Said, Nabil M. Hamed, Almoataz Y. Abdelaziz. (2016). Shuffled Frog Leaping Algorithm for Economic Dispatch with Valve Loading Effect. *International Electrical Engineering Journal, Vol 7 No 3*
- [11]