http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (6): 1014-1024, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.1014.1024

Study on Motion Planning of a Three Limb Robot

[1]Ming Pang, [1]Han L. Yang, [1]Tie F. Zhang and [2]Jie Zhao
[1]College of Light Industry, Harbin University of Commerce, Heilongjiang, China
[2]Robot Research Institute, Harbin Institute of Technology, Heilongjiang, China

Abstract: A novel three limb robot was described and two motion planning methods were discussed. After the introduction of the robot mechanical structure and the human-robot interface, method A is on the basis of the genetic algorithm. A kind of variable structure genetic algorithm is proposed in method A to solve the problem of motion planning of the three limb in dynamic environments. The variable structure genetic algorithm changes the original structure by abandoning Elitist Model, expectation selection, reproducing population and changing the probability of crossover and mutation. Experiments results show that the former algorithm is effective in static environments and the latter algorithm is good at dynamic environments. Method B is a two-grade search mechanism. The first-grade search method using genetic algorithm tries to find an optimized target position and orientation of the three limb robot. The second-grade search method using virtual compliance tries to avoid the collision between the three limb robot and obstacles in a dynamic environment. Experiment shows the feasibility of the two-grade search mechanism and proves that the proposed motion planning method can be used to solve the motion planning problem of the redundant three limb robot without deficiencies of traditional genetic algorithm.

Key words: Three limb robot, motion planning, genetic algorithm, two-grade search mechanism

INTRODUCTION

An increasing interest in the development of special climbing robots has been witnessed in last decades. The motivations behind it are to increase operation efficiency and protect human health and safety in dangerous tasks, such as cleaning high-rise buildings, spray painting and sand blasting of gas tanks, inspecting and maintaining nuclear facilities. Climbing robots, with their capabilities to adhere to wall surfaces and move around carrying appropriate sensor or tools, are able to replace human workers in these dangerous duties and eliminate costly erection scaffolding (Yan *et al*., 1999).

And at the same time, climbing robots should have the operation capability. Japanese researcher professor Noriho Koyachi developed a new concept of limb structure of walking robots (Tummala *et al*., 2002). The major Characteristic of limb structure is that the “limb” of a robot can be used to both walk and operate. This study introduces such a three limb robot.

Three limb robot is a joint type robot which is difficult to plan its motion in dynamic spaces. The problem of motion planning with obstacle avoidance has been extensively studied over the last decade. The main task of motion planning for robot end effectors is to find an collision-free trajectory from an initial to a final configuration. Koyachi *et al*. (1993) presented a heuristic hierarchical search method for an industrial robot with 6 Degree of Freedom (DOF). The collisions are detected in the Cartesian workspace by a hierarchical distance computation based on the given CAD model which is done by adjusting the step size of the search to the distance between the robot and the obstacle. Recently, Genetic Algorithms (GAs) have been applied to robot path and motion planning problems. Tian and Collins (2004) applied a genetic algorithm to solve the position and movement of an end effector on the tip of a two joint robot arm. He defined objective functions in both Cartesian space and joint space and combined them to optimize the robot trajectory. Optimum solutions with smooth trajectories and minimal joint rotation were obtained. De La Cueva and Ramos (1998) proposed a simple method based on a genetic algorithm, where a polynomial approximates time histories of the trajectory in joint space. The genetic algorithm determines the parameters of the polynomial to minimize the fitness of the objective function. Pack developed a method to search for valid solution in configuration space based on genetic algorithm. He formulated the trajectory planning problem with point obstacles. His method can also be extended to an n-dimensional space. Jiao and Wang (2000) also proposed a genetic algorithm based trajectory planning method for a two degree of freedom robot manipulator whose workspace includes several point obstacles.

Corresponding Author: Ming Pang, College of Light Industry, Harbin University of Commerce, Heilongjiang, China

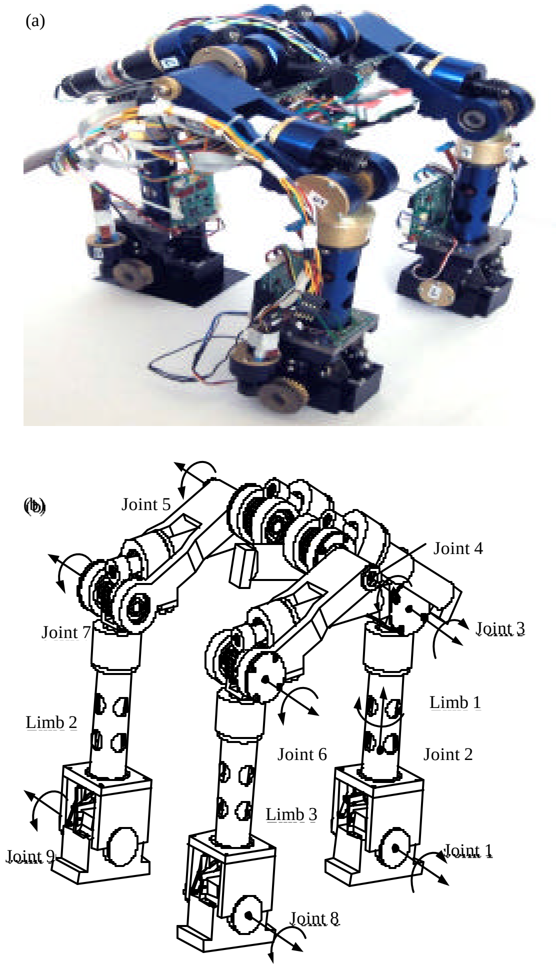

Fig. 1: Three limb robot with suction cups prototype

Two motion planning methods are proposed in this study. Method A is a novel motion and trajectory planning method for a three-limed robot's end effectors. Unlike the robots that referred by the researchers above (their joint hypothetically can rotate from 0 to 2π), the motion planning of the three limb robot is more difficult because of the restraint of its joint. Especially in the dynamic environments this will bring abortion of the almost all individual in the current population. Further more, the algorithm can easily stop or oscillate at a local pole. The proposed variable structure genetic algorithm can be used to solve the problem and find a feasible planning finally.

Method B is a two-grade method. The first grade of this method is to search a feasible objective position and orientation. The second grade is virtual compliance method which is used to solve the problem of how to move from initial to objective position and orientation in a complex environment.

Mechanical structure: The purpose of the three limb robot is semi-autonomous reconnaissance in dynamic and unstructured environments. We choose a mechanical structure illustrated in Fig. 1 with nine joints driven by nine motors so that, the robot can walk and operate flexibly.

The dimension of the prototype robot is approximately 240 mm in height, 230 mm in width. The robot weight, without onboard hardware, is approximately 7700 g.

The simulation experiments in this paper are realized in the human-robot interface based on Java/Java3D develop platform. This interface integrates the command user interfaces, mission planner, motion planner, trajectory planner, dynamics algorithm and 3D virtual environment in one which make operator easily control and supervise the robot. The human-robot interface based on Java/Java3D developed platform is shown in Fig. 2.

METHOD A OF MOTION PLANNING OF THE ROBOT

The first method of motion planning of three limb robot based on progressive genetic algorithm 1>:

Progressive genetic algorithm configuration description: Define the configuration as follow:

$$C_i = [\theta_{i1}, \theta_{i2}, \cdots, \theta_{im}] \quad i = 1 \cdots, m \tag{1}$$

Where, n is the number of robot joints and m is the total number of the amount of robot configuration in a path. Then the path can be defined as:

$$p = \{C_1, C_2, \cdots, C_m\} \tag{2}$$

Robot path planning is defined as follow:

- Robot should move along the "p" path to arrive at the aim point
- Impliedly, C_i can represent the configuration of robot at any time
- If the "p" path is accordant with the two above, then it is a feasible path
- If the "p" path is accordant with the optimization rule such as shortest path or shortest time, then it is a optimization path

Parameter coding: GA is a search engine based on neighborhood concept. We have the definition: on the

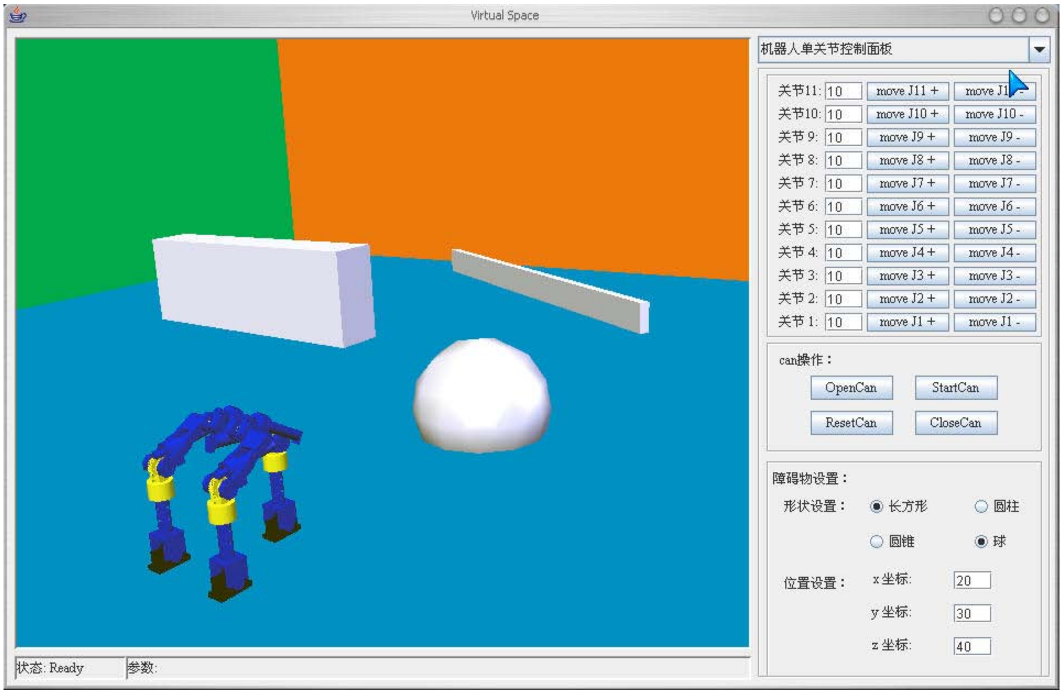

Fig. 2: Human-robot interface of the three limb robot

assumption that C_1 and C_2 is two of the robot configuration, where $C_1 = [\theta_{11}, \theta_{12},\cdots, \theta_{in}]$, $C_2 = [\theta_{21}, \theta_{22},\cdots, \theta_{2n}]$. If a given angle ε is accordant with the condition $\{|\theta_{1i}-\theta_{2i}|\}<\varepsilon$ for all the element of the sets $\{(\theta_{11}-\theta_{21}), (\theta_{12}-\theta_{22}),\cdots, (\theta_{1n}-\theta_{2n})\}$, then C_1 and C_2 is neighborhood each other. If predefine a fixed angle ε which is the maximum angle movement of any joint from one configuration to other, then any configuration can be represented as $[S_1, S_2,\cdots, S_n]$.

Where $S_i \in \{-2, 0, +2\}$ represent the rotation direction of joint i. So, a completed path of robot can be represented as:

$$[S_{11}, S_{12},\cdots, S_{1n}], [S_{21}, S_{22},\cdots, S_{2n}], [S_{m1}, S_{m2},\cdots, S_{mn}] \quad (3)$$

Each individual of GA population is represented by the vector 3. When $\varepsilon \rightarrow 0$ and $m \rightarrow \infty$, vector 3 will steer the robot to the aim position.

Fitness evaluation: Genetic algorithm is an optimization method with multi constrained condition. Fitness function is essentially objective function in optimization problem. The optimization trajectory should be accordant with the rule as follow:

$$\text{Fitness} = C_{max}-d \quad (4)$$

where, d is the distance of the initial position and target position of robot end effectors; C_{max} is a positive constant.

Operator genetic algorithm: Firstly fitness proportion method is used to realize reproduction. According to fitness function, reproduction probability is gotten which is used to decide the number of offspring of current individual. The individual that has a bigger copy probability would have more offspring. The individual that has smaller copy probability may be eliminated. Then decide the crossover probability P_C and construct the matching pool in term of P_C. The individual in the matching pool is matched randomly. The position of crossover is also decided randomly. Finally, the individual has the opportunity of the probability P_C to mutate.

Implement of genetic algorithm: Thus the step of the genetic algorithm is as follow:

- **Step 1:** Decide the range of each factor and the length of the chromosome code
- **Step 2:** Randomly produce n individual to construct the initial population P(0)
- **Step 3:** Decode each individual and get the fitness of each individual

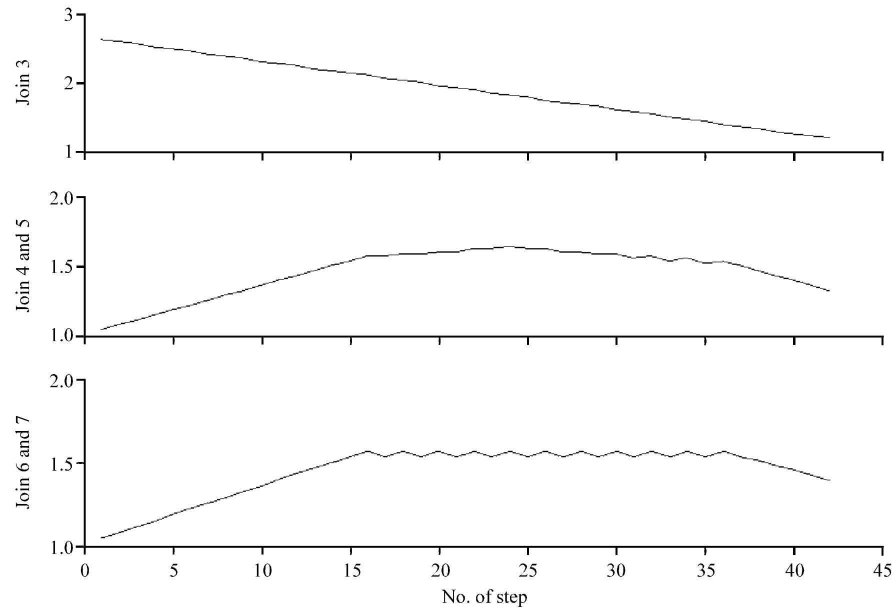

Fig. 3(a-c): Joint trajectory of, (a) 60 and 70, (b) 4 and 5, (c) 3 computed by the genetic algorithm

- **Step 4:** Operate the population P(t) according to the reproduction, crossover and mutation operator, then produce the population P(t+1)
- **Step 5:** Repeat the step 3 and step 4 until the current phase target is achieved or approached
- **Step6:** Repeat the step 1~5 until the final position is achieved

According to the algorithm above, the joint trajectory is gotten as in Fig. 3.

Joint 3, 4, 5, 6 and 7 are used to realize this motion. Our objective is to let the robot end effectors attain the final position. Figure 4 shows that the initial and final position and orientation of the three limb robot.

Variable structure genetic algorithm: Now let us suppose that an obstacle appear and the robot may collide with the obstacle. The results using the genetic algorithm above are shown as in Fig. 5 and 6.

In Fig. 5 we can see the latter trajectory is an oscillation at a local pole. This is because the individual in the latter population are all aborted. To solve the problem the variable structure genetic algorithm is developed here. The algorithm changes the original structure by abandoning Elitist model, expectation selection, reproducing population and changing the probability of crossover and mutation.

Fitness evaluation: Fitness function can be defined as follow:

$$\text{Fitness} = C_{max} - \sum_{i=1}^{4} w_i f_i \tag{5}$$

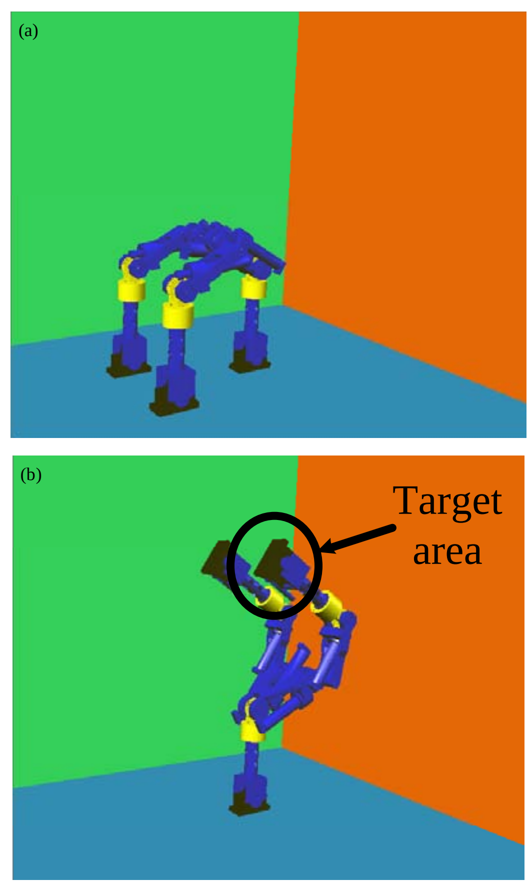

Fig. 4(a-b): Initial and final position and orientation

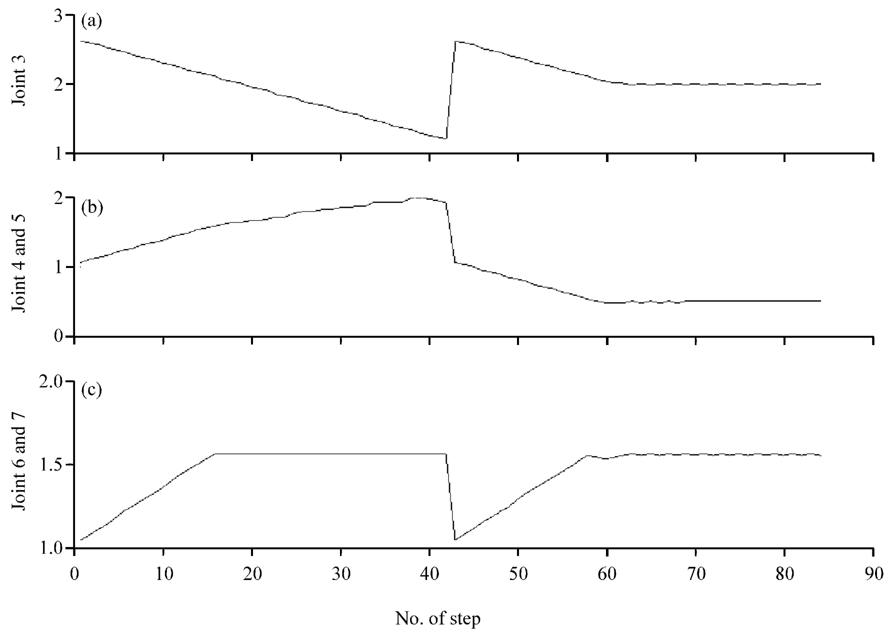

Fig. 5(a-c): Joint trajectory of (a) 6 and 7, (b) 4 and 5, (c) 13 computed by when robot

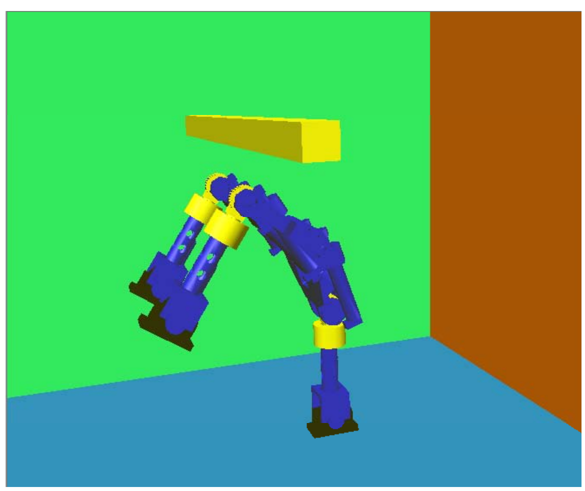

Fig. 6: Robot oscillate at the obstacle area

where, C_{max} is a positive constant; $f_i \in \{0,1\}$ represents the effectiveness of the current configuration; f_2 represents the amount of collision between robot and obstacle; f_3 represents the distance between robot and objective; f_4 is the step length. In the process of population evolvement, the individual with higher fitness would have more chance to participate in the competition in the next generation; the individual with lower fitness would be eliminated gradually.

Elitist model and reproduce population: In the genetic algorithm above Elitist model will provide the motion direction until the robot gets to the target area. In dynamic environments Elitist model will bring forth an oscillation at a local pole. So, in such a generation Elitist model should be abandoned. In addition, because of the restraint of joints and existence of some obstacles almost all the individual abort. The current population will be replaced by a new population produced randomly.

Expectation selection method: Expectation selection method is shown as follow:

- Compute the expectation $\bar{f}_i$ of the fitness:

$$\bar{f}_i = \frac{1}{n}\sum_{i=1}^{n} f_i \qquad (6)$$

- Compute the expectation of each individual in the population:

$$\bar{R}_i = \frac{f_i}{\bar{f}_i} \qquad (7)$$

Adjust the probability of crossover and mutation: The probability of selection and mutation can be adjusted adaptively according to the formulation as follow:

$$P_c = P_{c0} - \frac{f^i_{t,c} - f_{t,min}}{f_{t,max} - f_{t,min}} \cdot \frac{1}{1+e^{-k_c \Delta_i}} \qquad (8)$$

$$P_m = P_{m0} - \frac{f^i_{t,m} - f_{t,min}}{f_{t,max} - f_{t,min}} \cdot \frac{1}{1+e^{-k_m \Delta_i}} \qquad (9)$$

Where:

$f_{t,c}$ = The bigger fitness one between the two crossover individual
$f_{t,m}$ = The fitness of the individual that are going to mutate
P_C = Crossover probability
P_m = Mutation probability
P_{C0} = Initial crossover probability
P_{m0} = Initial mutation probability
k_C = Constant decided by experiments
k_{Cm} = Constant decided by experiments

RESULTS OF METHOD

In allusion to the environment referred above we used the proposed variable structure genetic algorithm to plan the robot's motion. Figure 7 is the joints trajectory and Fig. 8 is the final position and orientation of the three limb robot.

Experiments results show that the variable structure genetic algorithm can solve the motion planning problem in dynamic environments. The joints trajectory has slim oscillation in some phase seen from Fig. 8. These oscillation can be eliminated easily.

METHOD B OF MOTION PLANNING OF THE ROBOT

Two grade algorithm of planning of the three limb robot

Problem description: We suppose that the robot wants to climb the wall from the ground (the initial position of the robot is on the ground and there are some obstacles in robot's motion space). Then how to autonomously get the objective position and orientation with one or two feets on the wall, especially how to move from the initial

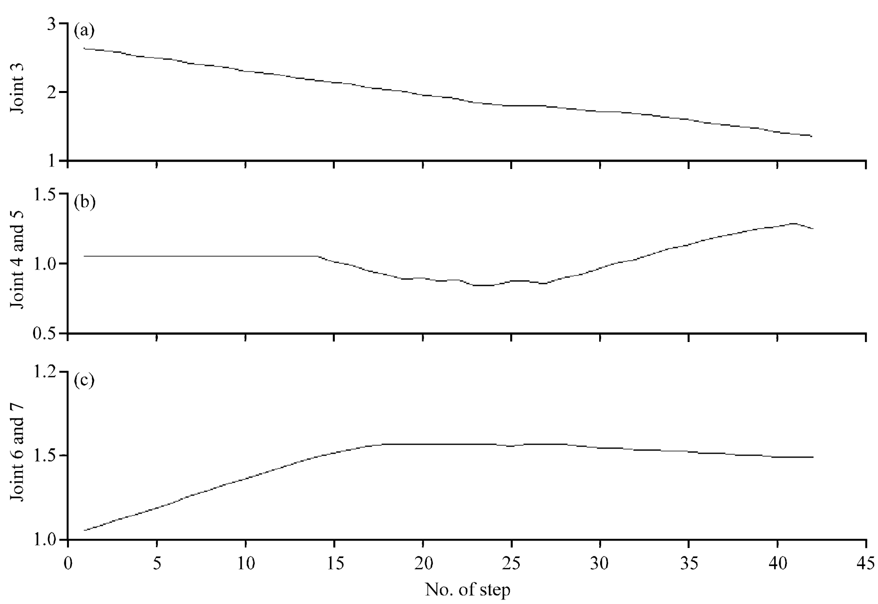

Fig. 7(a-c): Joints trajectory of, (a) 6 and 7, (b) 4 and 5, (c) 3 computed by variable structure GA

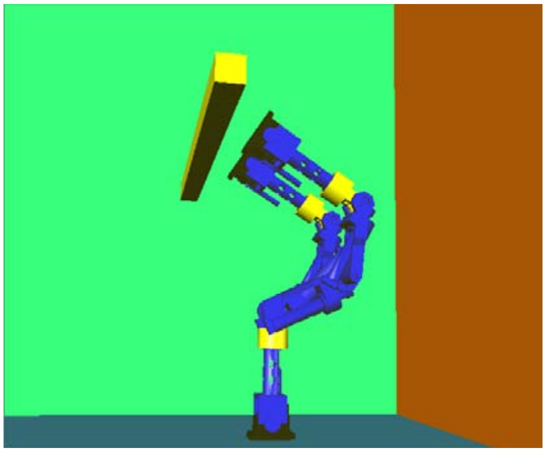

Fig. 8: Final position and orientation

position and orientation to the final one (the objective) is the problems that the motion planning method mentioned here must solve.

Configuration description: Define the configuration as follows:

$$c_i = [\theta_{i1}, \theta_{i2}, \cdots, \theta_{in}] \quad i = 1, \cdots, m \tag{10}$$

where, n is the number of robot joints and m is the total number of robot configuration in a path. Then the path can be defined as:

$$P = \{c_1, c_2, \cdots, c_m\} \tag{11}$$

Robot path planning is defined as follows:

- Robot should move along the "p" path to arrive at the aim point
- Impliedly, c_i can represent the configuration of robot at any time
- If the "p" path is accordant with the two definitions above, then it is a feasible path
- If the "p" path is accordant with the optimization rule such as the shortest path or shortest time, then it is an optimization path

First grade algorithm of the planning method: The purpose of GA is to search a feasible objective position and orientation. GA is an optimization method with multi-constrained condition. Fitness function is essentially objective function in optimization problem. The optimization path should be consistent with the following rules:

- Avoiding ineffective configuration.
- Collision avoidance
- Minimum position error

Thus, fitness function can be defined as follows:

$$\text{Fitness} = C_{max} - \sum_{i=1}^{3} w_i f_i \tag{12}$$

where, C_{max} is a positive constant; $f_1 \in \{0,1\}$ represents the effectiveness of current configuration; f_2 represents the amount of collision between robot and obstacle; f_3 represents the distance between robot and objective. In the process of population evolvement, the individuals with higher fitness would have more chance to participate in the competition in the next generation; the individuals with lower fitness would be eliminated gradually.

The step of the first-level GA algorithm is as follows:

- **Step 1:** Randomly build initial population named POPU, each individual represents a robot configuration $c_f = [\theta_{f1}, \theta_{f2}, \cdots, \theta_{fn}]$ and the three feet's position coordinate of any robot configuration is changeless
- **Step 2:** Eliminate those robot configuration individuals that contact with or drill through some obstacles
- **Step 3:** Rebuild the second generation by reproduction, cross-over and mutation
- **Step 4:** Elitist model is required. If the elitist individual accords with precision requirement then stop evolvement and reserve this individual to the NEW POPU array which stores the individual of the next population. If the elitist individual does not accord with precision requirement then repeat steps 2th~4th. The amount of the recurrent is at most 50. If the 50 generation evolvement process is completed and no individual accords with precision requirement then put the elitist individual in the last generation into the NEW POPU array
- **Step 5:** The NEW POPU array is the initial population of the next GA when it is full. Thus rebuild the second generation by reproduction, cross-over and mutation
- **Step 6:** After 100th generation, record the elitist individual

The result achieved by the algorithm above is the final target configuration of the three limb robot. Figure 9 illustrates the change process of fitness. Then the objective position and orientation are already achieved. Figure 10 shows respectively the initial and objective position and orientation.

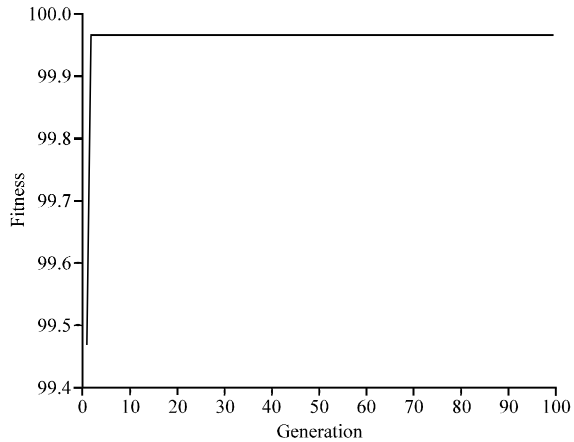

Fig. 9: Fitness changes process of GA

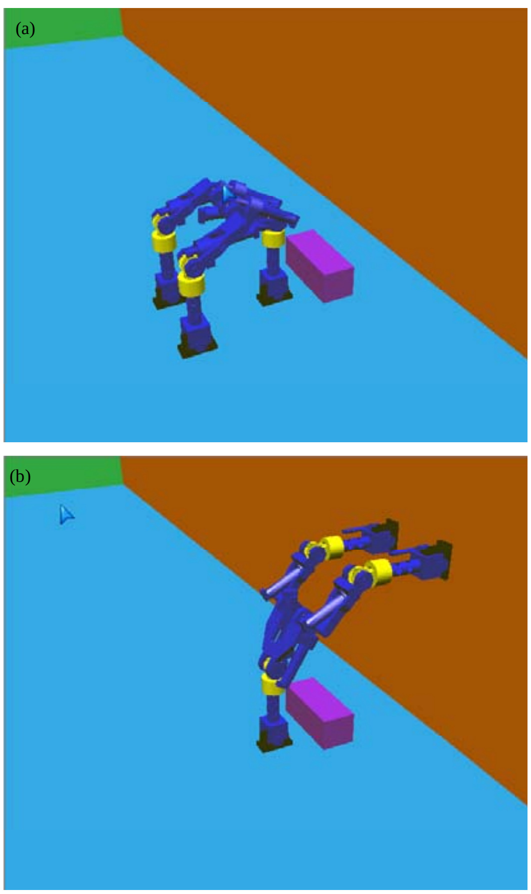

Fig. 10(a-b): Initial and objective position and orientation

Second grade algorithm of the planning method: Cubic splines interpolation method can be used to compute the joint trajectory from the initial to the objective position and orientation. Then the robot joint would rotate according to these joint trajectories. It is supposed that there is a new obstacle in the path that the robot wants to move along, thus robot may use the virtual compliance method to avoid the conflict with the new obstacle.

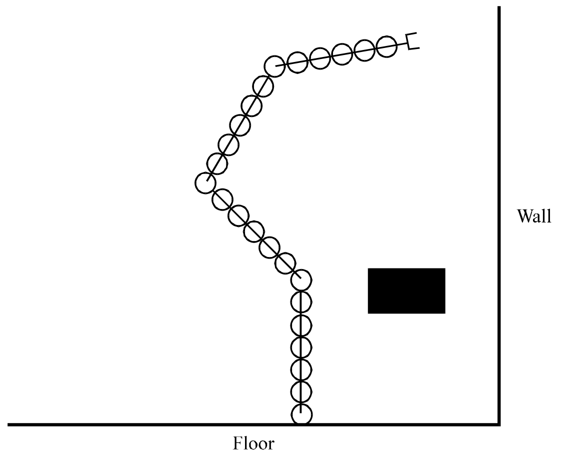

Fig. 11: Approximate contact circle of robot' body

In order to detect the collision between the robot's body and obstacles, the robot's body can be regarded as contact circle (sphere in 3D environment). For some given robot, (Beaumont and Crowder, 1991) defined the radius of one circle (sphere), number and position of the circles. If the distance between the robot's body and the obstacles is less than the radius of the defined circle, then collision happens. Figure 11 shows the circle of a robot arm's body.

Because the three limb robot's joint is composed of worm and worm gear, these nine joints are self-locked. It is difficult for the traditional compliance to be used in this type of robot. Compared with real compliance, virtual compliance does not deal with the problem of force/torque control and there is no contact between the obstacles and robot's body when the method works. Thus the impact that may act on the robot's body can be decreased to the lowest level.

The virtual compliance method is composed of a fuzzy controller, including fuzzification process, fuzzy reasoning mechanism and defuzzification process.

Fuzzification process is a process which transforms the input into fuzzy language variables by membership function. This process is the precondition of fuzzy reasoning. In this study, the fuzzy reasoning is composed of a set of IF-THEN statements. According to these fuzzy rules, a feasible solution can be achieved in output space. Defuzzification process is a process which transforms the fuzzy solution into a clear solution.

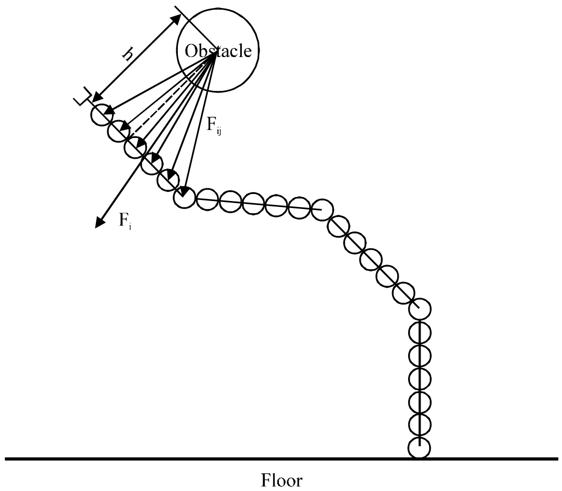

Fig. 12: Virtual resultant repulsive force

The input of the fuzzy logic system is the virtual repulsive torque applied by obstacles to any bar of the three limb robot. The virtual repulsive torque's calculation is as follows:

$$T = F \cdot h \tag{13}$$

where, T is the virtual repulsive torque, F is the virtual resultant repulsive force and h is the distance between the center of an obstacle and the bar of the three limb robot.

The virtual resultant repulsive force is the composition of all virtual forces applied by obstacles to each contact circle in a bar of the three limb robot. Figure 12 shows the virtual resultant repulsive force.

The virtual resultant repulsive force's calculation is as follows:

$$F_i = \sum_{j=1}^{6} F_{ij} \tag{14}$$

where, F_i represents the resultant repulsive force on the ith bar, F_{ij} is the repulsive force on the jth contact circle in the i bar. The number i here is from 1 to 3. The calculation of the repulsive force F_{ij} is as follows:

$$F_{ij} = \frac{1}{d^{\alpha}} \tag{15}$$

where, a is an constant greater than 1, here it is equal to 2; The distance d is between the centres of the contact circles (or spheres) and the robot's body bar. It measure in millimetre. The output is the adjusted angle values of the joint 3, 4, 5, 6 and 7. The set of fuzzy rules is described as:

$$R = \{R_1, R_2, \cdots R_n\} \tag{16}$$

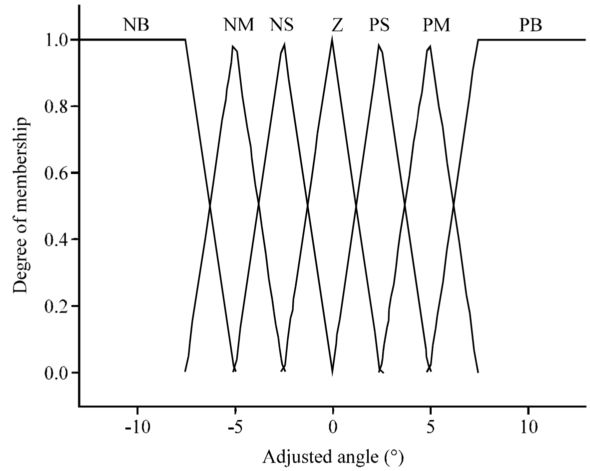

Fig. 13: Membership functions of output

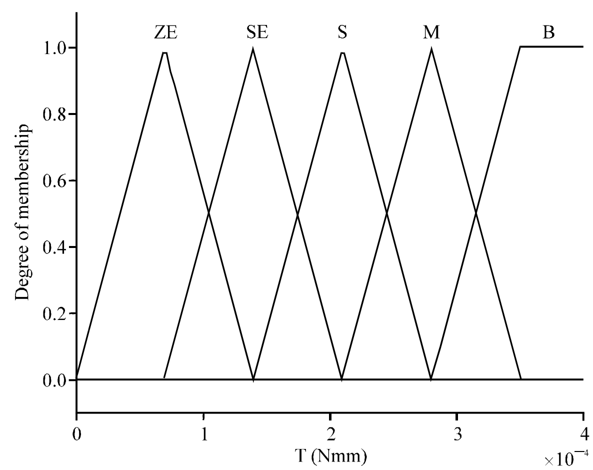

Fig. 14: Membership functions of input

On account of space limitation, only three of the rules are described as follows:

R_1: IF T_1 = ZE and T_2 = ZE AND T_3 = B, THEN $\Delta\theta_3$ = MS and $\Delta\theta_4$ = $\Delta\theta_5$ = Z and $\Delta\theta_6$ = $\Delta\theta_7$ = Z

R_2: IF T_1 = ZE and T_2 = S and T_3 = ZE, THEN $\Delta\theta_3$ = MS and $\Delta\theta_4$ = $\Delta\theta_5$ = MB and $\Delta\theta_6$ = $\Delta\theta_7$ = Z

R_3: IF T_1 = B and T_2 = ZE and T_3 = ZE, THEN $\Delta\theta_3$ = Z and $\Delta\theta_4$ = $\Delta\theta_5$ = MS and $\Delta\theta_6$ = $\Delta\theta_7$ = MB

The fuzzy sets for the output are selected as negative-big (NB), negative-medium (NM), negative-small (NS), negative-zero (NZ), zero (ZE), positive-zero (PZ), positive-small (PS), positive-medium (PM), positive-big (PB). The corresponding membership functions are selected as shown in Fig. 13. Similarly, the fuzzy sets of the input are ZE, smaller (SE), small (S), medium (M) and big (B). The membership functions for the fuzzy sets are selected as shown in Fig. 14.

Experiment: According to the two-grade method mentioned above, an experiment is implemented with the three limb robot to climb the wall in a dynamic environment. The three limb robot stands on a level ground made of a steel plane and she attempts to climb the wall made of a steel plate too.

First a feasible path is produced by GA in human-robot interface. Then the joint trajectories are computed by cubic spline interpolation. The human-robot interface can control the virtual and real three limb robot simultaneously. This is due to the reason that the human-robot interface transmits the movement information to the three limb robot after the virtual robot moves by one step length and then the real robot follows the virtual robot's motion. In the process of the three limb robot's movement along the trajectories, there is a sudden obstacle appearing over the robot. This information can be transmitted to the human-robot interface real time. Virtual compliance method is activated at this time to avoid the conflict between the robot and the obstacle. Finally the robot arrives the objective position and orientation.

Figure 15 shows the joint trajectories of the three limb robot. The motion is divided into three phases. The first grade algorithm is used in the first phase. The second grade algorithm, namely virtual compliance method, is applied to avoid collision in the second phase. In the third phase, obstacles are avoided and the first grade algorithm is used to complete robot's motion. Figure 16 shows the real and virtual three limb robot's motions.

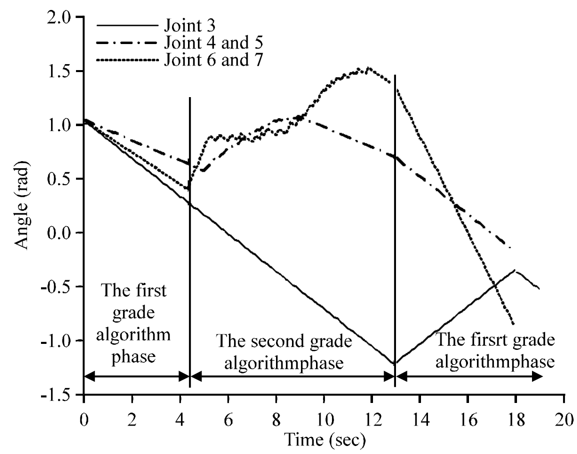

Fig. 15: Joint trajectories of the three limb robot using tow-grade search mechanism

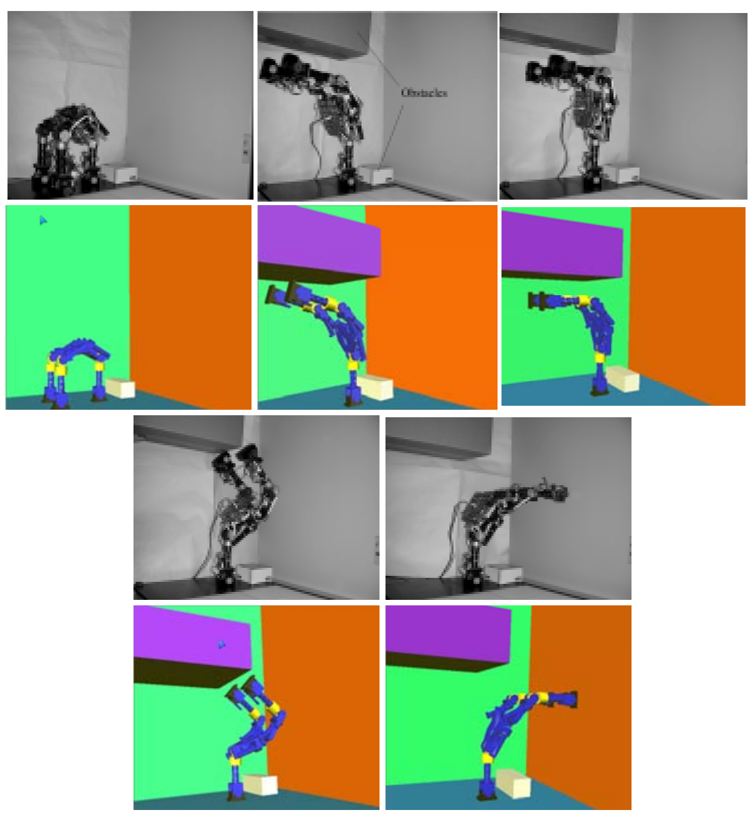

Fig. 16: Three limb robot's motion controlled by human-robot interface

CONCLUSION

From the present study the following conclusion have learn made:

- This study proposed a novel three limb robot with nine joints driven by nine motors in order to both walk and operate flexibly
- A novel genetic algorithm is developed for the motion planning of the three limb robot. Simulation example shows the feasibility of the genetic algorithm. And the variable structure genetic algorithm is proposed to realize motion planning of the three limb robot in dynamic environments. Experiments results show the validity of this kind of genetic algorithm
- Two-grade motion planning method is proposed for the three limb robot. The first grade method using GA tries to find an optimized target position and orientation of the three limb robot. The second grade method tries to deal with the motion of the three limb robot in dynamic environment. The two-grade search mechanism method can fill up the deficiency of GA in motion planning of the three limb robot. Experimental result shows the feasibility of the two-grade motion planning with satisfaction. The shortcoming of the two-grade motion planning is that it is time-consuming to transmit the information of dynamic environment to the human-robot interface. The next study is to find a better way to quickly get the transmission used in the second-level GA.

REFERENCES

Beaumont, R.G. and R.M. Crowder, 1991. Real-time collision avoidance in tow-Armed robotic systems. Comput. Aided Eng. J., 8: 233-240.

De la Cueva, V. and F. Ramos, 1998. Cooperative genetic algorithms: A new approach to solve the path planning problem for cooperative robotic manipulators sharing the same work space. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Volume 1, October 13-17, 1998, Victoria, BC., Canada, pp: 267-272.

Jiao, L. and L. Wang, 2000. A novel genetic algorithm based on immunity. IEEE Trans. Syst. Man Cybernet. Part: A Syst. Hum., 30: 552-561.

Koyachi, N., T. Arai, H. Adachi and Y. Itoh, 1993. Integrated limb mechanism of manipulation and locomotion for dismantling robot-basis concept for control and mechanism. Proceedings of the IEEE/Tsukuba International Workshop on Advanced Robotics, November 8-9, 1993, Tsukuba, Japan, pp: 81-84.

Tian, L. and C. Collins, 2004. An effective robot trajectory planning method using a genetic algorithm. Mechatronics, 14: 455-470.

Tummala, R.L., R. Mukherjee, N. Xi, D. Aslam and H. Dulimarat *et al*., 2002. Climbing the walls. IEEE Robot. Autom. Mag., 9: 10-19.

Yan, W., L. Shuliang, X. Dianguo, Z. Yanzheng, S. Hao and G. Xueshan, 1999. Development and application of wall-climbing robots. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 2, May 10-15, 1999, Detroit, MI., USA., pp: 1207-1212.