PATH PLANNING OF COOPERATIVE CONSTRUCTION MANIPULATORS USING GENETIC ALGORITHMS

PL. Sivakumar*, Koshy Varghese and N. Ramesh Babu*****

**Research Scholar, Dept. of Civil Eng., I.I.T. Madras, India*
***Asst. Prof., Dept. of Civil Eng., I.I.T. Madras, India* – koshy@civil.iitm.ernet.in
****Asso. Prof., Dept. of Mech. Eng., I.I.T. Madras, India*

Abstract: This paper presents the work done to investigate the potential of applying Genetic Algorithms (GA's) to automate the path planning of cooperative construction manipulators. Though there is potential to achieve economy by using cooperative manipulators, this method is not being commonly adopted in construction sites. A key barrier to the use of cooperative manipulators is the meticulous planning required to ensure feasible and safe operation. The basic premise of this work is that automating the different planning steps will contribute to more reliable plans and thus promote the usage of cooperative manipulator. Two methodologies have been proposed using the concept of Configuration Space (C-Space) technique in conjunction with the genetic search. The implementation details and results of the best methodology is presented and discussed for a test problem involving two cooperative manipulators each of 2 DOF.

Key Words: Cooperative Construction Manipulator, Path Planning, Genetic Algorithms

1. INTRODUCTION

Investigation on the applications of automation and robotics is an active area of research work in construction. The basic objectives behind these investigations are (a) To speed up the construction operations and thereby to achieve better productivity (b) To arrive at better plans for construction operations in terms of safety and quality (c) To minimize the cost of the construction operations. The concept of automation can be applied to any of the following stages in a construction project: planning, execution, maintenance or rehabilitation.

This paper presents the work done to investigate the potential of applying GA's for path planning of cooperative construction manipulators. Construction manipulators can take the shape of specialized equipment such as a pipe manipulator or common equipment such as a crane. Cooperative manipulators are becoming essential in construction situations. For example, utilization of two medium capacity cranes (which are normally available at construction sites) to cooperatively lift a heavy object will be more economical than using a large capacity crane or jacking systems.

For all types of manipulators, manual planning is cumbersome because of the limitations in visualization and difficulty in evaluating alternatives for different combinations of input parameters. For example, modeling the kinematics of the equipment, checking for interference and representation of space (which are required to automate the path planning procedures of two cooperative crane manipulators) are not suited for manual analysis. The automation of path planning tasks will reduce the chances of human error, increase planning productivity and allow the planners to evaluate alternate plans and thereby to arrive at a better plan.

The need and applicability of large scale construction manipulators for different activities such as piping erection, elevated concrete placement, painting, scaffolding erection and demolition in the construction environment has been identified by Hsieh [4].

Past work has also identified the importance of path-planning of construction manipulators. A path-planner based on dynamic programming concepts was developed for a six DOF pipe manipulator [1]. A system called Path-finder was developed for a generic construction manipulator [8]. Path planning method of single crane lifts has been automated using C-Space concepts in conjunction with heuristic search [10].

Path Planning is a well established area in Robotics. A description of basic issues involved in path planning and different approaches to solve them have been identified [5, 11]. C-Space, essentially a representational tool, is an imaginary space where the robot manipulator is represented as a point, surrounded by space represented by its degrees of freedom. C-Space concept is extensively used to solve path planning problems. This idea of shrinking a robot manipulator to a point was introduced by Udupa [12] and later it was systematically analyzed and popularized by Lozano-Perez [6].

GA has been extensively applied to solve path/trajectory planning problems in robotics field because of its (1) Robustness to solve a wide range of optimization problems in different fields of application (2) Ability to customize and design genetic operators to suit the problem requirements (3) Suitability to parallel computing environment (because of the repetitive functional evaluations) to tackle path planning problems with high computational complexity. A fundamental coverage of GA's has been presented by Goldberg [3]. A number of researchers have worked on designing special GA's which makes GA's more close to natural systems by capturing key features (-such as Co-evolution and Life Time Fitness Evaluation) subjected to problem requirements [9]. Customized genetic operator such as analogous crossover which suit certain problem situations have also been designed [2]. Parallel GA's were developed to tackle path-planning problems of high computational complexity [7].

There are differences in the criteria used to assess the suitability of a path in conventional robotics and construction situations. This is because of the key requirements of a crane manipulator such as telescoping, heavy loading and hoisting operation. So, conventional path planning techniques, applied in robotics field, can not be directly applied to construction manipulators [13]. In addition, using conventional search techniques require several hours of execution time to determine a path for a 3 DOF crane within a realistic model of lift area [10]. As a result, the performance of the system will be unacceptable for problems involving multiple manipulators. Based on this, it was decided to investigate alternate approaches such as GA's for path planning problems involving multiple cooperative construction manipulators.

This paper consists of six sections. The next section presents the details of a test problem considered for implementation. The third section focuses on the concepts and solution methodologies to solve the problem. Two methodologies have been proposed to solve the test problem. The implementation details of the best methodology is presented in the fourth section. The fifth section presents observations made while solving the test problem. Finally conclusions are presented in the sixth section.

2. PROBLEM STATEMENT

This section presents the details about a simple path planning problem involving two planar cooperative manipulators each of 2 DOF. It also presents the key features to be considered to formulate the problem.

The primary objective of solving this simple test problem is to identify the difficulties that may arise due to the cooperation of the two manipulators and devising methodologies to tackle the same. Figure 1 shows a cooperative manipulator system involving two manipulators each of two DOF. From hereafter this problem is termed as Planar 2x2 Manipulator Problem.

Note: Planar n x m Manipulator Problem refers to a path planning problem which involves 'n' manipulators each of 'm' DOF. A cooperative manipulator system refers to an assembly of two manipulators with the object handled by them.

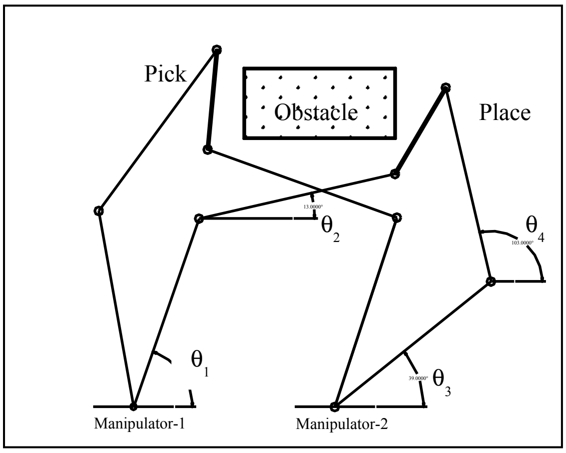

Figure 1. Planar 2x2 Manipulator

Key features to be considered in the above cooperative manipulator system

Validity

This condition refers to the capability of the two manipulators to handle the object cooperatively at an unique position of the cooperative manipulator system. This is ensured by checking whether the distance between the end effectors of the two manipulators is equal to the object length subjected to a permissible limit.

Interference

There should be no collision in the cooperative manipulator system. Collision in the cooperative manipulator system may be due to any one of the following cases: (a) Manipulator-1/manipulator-2 to obstacle (b) Object to obstacle (c) Manipulator-1 to manipulator-2.

Valid sequence limit

Each path is represented by an array of definite number of successive configuration sets. Each configuration set i.e. $[\theta_1,\theta_2,\theta_3,\theta_4]$ represents an unique configuration of the cooperative manipulator system. The difference in joint angle rotations between successive steps at each joint should not exceed a specified limit. This condition ensures that there is no collision in the cooperative manipulator system as it moves between successive steps. This limit is fixed based on the minimum size of the obstacle.

Work done by the manipulator

An optimal path is the one that ensures the movement of the cooperative manipulator system from pick to place location with a minimum amount of work done. The criteria to compute minimum amount of work done will depend on the field of application. For example, in case of a crane manipulator, it depends on (1) Number of swing, luff and hoist operations (2) Magnitude of swing, luff and hoist operations (3) Proximity of the path to obstacles (4) Percentage of crane capacity utilized at each step (5) The way by which the load travels i.e. whether over the obstacle or over a plain area.

In this case, the work done by the manipulators is considered as the sum of joint angle rotations at all the joints between successive configuration sets as the cooperative manipulator system moves from pick to place location.

3. SOLUTION CONCEPTS AND METHODOLOGY

This section presents the search options to solve a cooperative manipulator path planning problem, two proposed methodologies to search in C-Space, details about C-Space computation and methods to automate C-Space computation .

3.1 Search Options

Basically, search required to solve a path planning problem can be done by two ways:

a. Searching in the real space.
b. Searching in the C-Space.

Search in the real space is recommended when there is no redundancy in the manipulator system. Here, redundancy refers to the existence of more than one configuration of the manipulator for the same position of end effector (single manipulator)/object (cooperative manipulator system).

Search in the C-Space is appropriate when there is redundancy in the manipulator. C-Space is an imaginary space having the number of dimensions equal to the degrees of freedom of the manipulator. So, each point in the C-Space represents an unique configuration of the manipulator. The basic idea of C-Space is to represent the manipulator as a point. This mapping transforms the problem of planning the motion of a dimensioned object into a problem of planning the motion of a point [5].

3.2 Two Methodologies for Searching in C-Space

3.2.1 Methodology-1: Online Interference Check–Path String Approach

In this approach, the path is expressed in terms of the configuration parameters of the manipulator. Search is done in the total C-Space which includes both feasible and infeasible space. To evaluate the fitness of individuals at each generation, interference detection is performed in parallel to the GA iterations at each generation.

3.2.2 Methodology-2: Search in a Pre-Computed Feasible C-Space

Feasible C-Space is first computed in this approach by evaluating all possible combinations for a configuration set. Genetic search is carried out only in the feasible C-Space i.e. GA is allowed to generate only those configuration sets which are identified as valid and collision free. This improves the effectiveness of the search significantly.

3.3 C-Space Computation

Configuration Space Computation refers to the computation of either obstructed or feasible C-Space. Basic techniques such as obstacle growth will be suitable to generate C-Space only for simple problems such as planning the path of a polygon in a two dimensional environment. The use of interference detection routines was found to be more suitable for path planning problems involving manipulators such as a 3 DOF single crane manipulator or Planar 2x2 Manipulator (as shown in Figure 1).

Computing feasible C-Space or obstructed C-Space depends on the type of the problem. For path planning problems of single manipulators, feasible C-Space will be much higher than the obstructed C-Space subjected to the degree of obstacles. In such cases, first the obstructed C-Space is computed, then it is used for evaluating the fitness of each individual. In case of path planning problems involving cooperative manipulators, the feasible C-Space will be much less than the obstructed C-Space. This is because of the cooperation, required between the two manipulators. In such cases, the feasible C-Space is computed first and the search is carried out only in feasible C-Space.

3.4 Automating C-Space Computation

This computation identifies either feasible or obstructed C-Space by performing repeated interference detection for all possible positions of the manipulator (for all the combinations of the configuration set). For path planning problems involving 2 DOF single manipulator, the total number of evaluations will be much less subjected to the step size used for incrementing each joint angle parameter. In such cases, the C-Space can be generated manually using a graph sheet. However, for path planning problems involving manipulators of higher DOF, it is not possible to compute the C-Space manually. In such cases, C-Space computation is automated in the following two ways:

a. In AutoCAD environment using the programming languages such as AutoLISP or ARX.
b. Developing interference detection routines using computational geometry concepts.

The second method is more effective because of its more efficient usage of CPU and hence will have a lower computational time.

4. IMPLEMENTATION DETAILS METHODOLOGY-2

This section presents the details about the feasible C-Space computation, genetic modeling and results for the Planar 2x2 Manipulator problem shown in Figure 1.

4.1 Feasible C-Space computation

Feasible C-Space consists of a set of configuration sets which are valid and collision free. The dimension of this feasible C-Space is four (because four joint angle parameters are required to represent an unique configuration of the cooperative manipulator system). This space can not be visualized and it is represented as an Euclidean space of four dimension. Feasible C-Space is computed by exhaustively evaluating all possible configuration sets against validity and collision conditions. Let 'r', 's' and 'NL' be the range for the movement of each manipulator, step size used to vary each joint angle parameter and number of links in the cooperative manipulator system respectively. The total number of configuration sets to be evaluated (N) is calculated from the equation (1).

$$N = (r/s)^{NL} \quad (1)$$

For the test problem, shown in Figure 1, the total number of evaluations are 18,74,161 [s: 5 degrees; r: 180 degrees and NL: 4]. The total number valid and collision free configuration sets are 10,394.

4.2 Genetic Modeling

4.2.1 Path representation as a string

Each configuration set in the pre-computed feasible C-Space is assigned to an unique identification number (c-set id). Each path is represented as an array of definite number of configuration set id's. The number of steps required to represent a path depends on the feasible work envelope/space (accessibility of each arm of each manipulator), number of obstacles and the way in which the obstacles are distributed in the construction site. In this case, the number of steps between the pick and place location is specified by the planner as five. The path is expressed as follows:

PICK, $[\text{c-set id}]_1$, $[\text{c-set id}]_2$, $[\text{c-set id}]_3$, $[\text{c-set id}]_4$, $[\text{c-set id}]_5$, PLACE

To evaluate the fitness of the individual, the path in terms of configuration set id's is represented as a path in terms of configuration sets by referring to the pre-computed feasible C-Space.

4.2.2 Decoding schemes

The number of bits allocated to each c-set id depends on the total number of configuration sets representing the feasible C-Space. The total number of feasible configuration sets are 10,394 (from section 4.1). The equivalent binary number to 10,394 is 10100010011010. So a minimum of 14 bits are required for each c-set id. There will be a possibility of the decoded value of c-set id being more than 10394, since the decoded value is 16383 if all bits assume a value of '1'. So, it is necessary to map the decoded value of each c-set id from the range [0, 16383] to the range [0, 10394]. This is done by dividing all the decoded value of c-set id by the ratio 16383/10394.

The resolution of genetic search in this approach depends on the resolution of the pre-computed feasible C-Space.

4.2.3 Fitness

Objective function [f(x)]

The total work done by the cooperative manipulator system during its movement from pick to place location is to be minimized. It is equal to the sum of absolute differences of joint angles between successive steps at all the joints as the cooperative manipulator system moves from pick to place location. It is calculated from the equation (2).

$$f(x) = \sum_{i=1}^{n-1} \sum_{j=1}^{m} | \theta_{i+1, j} - \theta_{i, j} | \quad (2)$$

where 'n' is the number of the configuration sets representing a path, 'm' is the total number of joint parameters required to define an unique position of the cooperative manipulator system and '$\theta_{i, j}$' is the value of joint angle of j^{th} link in i^{th} configuration set.

Violation Coefficient [C]

The basic objective of this coefficient is to ensure that there is no collision in the movement of the cooperative manipulator system when it moves between two successive steps. This is achieved by limiting the angular rotation at each joint between two successive steps to a specified value. This limit is called as valid sequence limit. This limit is fixed based on the number of steps required to move from pick to place position and the least size of the obstacle. In this case, there is no need to check for validity and collision because all the configuration sets generated by GA are valid and collision free.

It is calculated as the ratio of number of joint angle rotations which exceeds the specified limit to the total number joint angle rotations. Total number of joint angle rotations is the product of (number of configuration sets representing the path, including pick and place sets–1) and number of joint angles per configuration set. The value of the valid sequence limit is fixed as 50 degrees in this case.

Fitness Function[F]

Fitness of each individual is calculated from the following equation.

$$F = f(x)\ [1+C] \qquad (3)$$

4.3 Results

Details of one of the best path generated by GA's is presented in the table 1. The movement of the cooperative manipulator system from pick to place location corresponding to the table 1 is shown in the figure 2.

Table 1. Details of path generated by GA's

Configuration at	Manipulator-1	Manipulator-2
Pick	[100,54]	[72,160]
Step-1	[120,55]	[75,165]
Step-2	[125,25]	[45,165]
Step-3	[95,15]	[30,165]
Step-4	[65,5]	[10,155]
Step-5	[70,0]	[5,100]
Place	[71,13]	[39, 103]

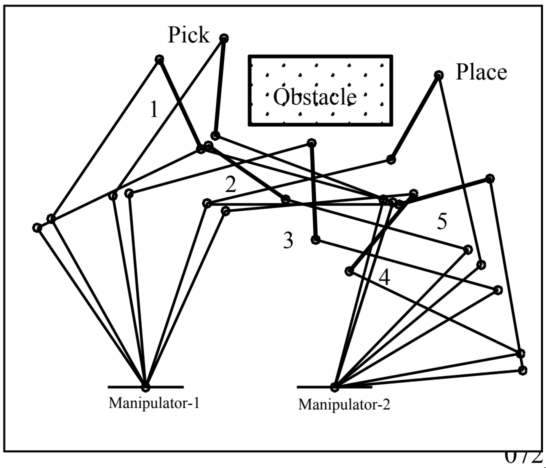

Figure 2. Path generated by GA's, in real space

However, for some of the runs, accumulation of steps on any one of the sides i.e. either near to the pick or place location was observed. As a result, there was a sudden jump between any of the two successive steps. This shows the significance of fitness function in the performance of GA's and the need for the improvement in the present formulation of fitness function.

Details of genetic parameters

Population size: 250; Probability of mutation: 0.001; Probability of crossover: 0.9; Maximum number of generations: 200; Crossover type: Two point crossover; Column convergence parameter: 0.8; Selection algorithm: Part sum method

5. OBSERVATIONS

This section illustrates the effectiveness of the proposed methodology for the Planar 2x2 Manipulator Problem.

5.1 Data Required

Problem specific data:

Total number of joint angles to represent an unique configuration: 4; Range for the movement of each link: 0 to 180 degrees; Step size: 5 degrees; Number of configuration sets representing a chromosome (path) excluding pick and place location: 5;

Data related to Genetic Modeling and C-Space Computation:

Number of bits allocated to each variable : 5.
Total possible number of values for each variable: 31 [$(11111)_2 = (31)_{10}$].
Step size used to discretize the range of each variable = 180/31 = 5.81.

Total number of combinations for each configuration set are 9,23,521 [$(31)^4$]. In case of genetic search without pre-computation, the number of bits allocated to each variable govern the total number of combinations for each configuration set.

Total number of valid and collision free configuration sets, identified from feasible C-Space computation, are around 10000 (with a step size of 5) and around 6000 (with a step size of 5.81).

5.2 Effectiveness of Search in a Pre-Computed Feasible C-Space

Size of the search space without pre-computation of feasible C-Space = $(9,23,521)^5 = 6.72 \times 10^{29}$. Size of the search space with pre-computation of feasible

C-Space $= [6,000]^5 = 7.78 \times 10^{18}$. From this, it is obvious that the size of the search space in a pre-computed feasible C-Space is less than 1% of what in case of without pre-computation of feasible C-Space.

5.3 Effectiveness of Search using Genetic Algorithms

Size of search space with prior computation in case of exhaustive search is 1.0×10^{20} $[(10,000)^5]$. Size of search space searched by GA's in a pre-computed feasible C-Space is 50, 000 [population size multiplied by maximum number of generations]. This is less than 0.1% of what in case of exhaustive search with prior computation of C-Space.

6. CONCLUSIONS

Based on the work presented in this paper, the following conclusions can be made:

1. C-Space technique in conjunction with search using GA's proves to be an effective approach to solve path planning problems of cooperative construction manipulators.

2. Utilization of computational geometry concepts to compute the feasible C-Space proves to be an effective approach when compared to the computation in AutoCAD environment using AutoLISP or ARX.

3. Search in a pre-computed feasible C-Space is much effective when compared to the search in the total C-Space. The size of the search space in case of search with pre-computation is less than one percent of what in case of without pre-computation.

4. Search using GA's is an effective approach when compared to other search techniques such as breadth first search and depth first search. The percentage of size of the search space searched by GA's is less than 0.1% of the space to be searched in case of exhaustive search.

On-going work is focused on investigating the applicability of this approach to more complex cooperative manipulator situations.

Acknowledgements

This work was supported by the Department of Science and Technology (DST), New Delhi, India, vide Grant No. III. 5(110)/97-ET.

REFERENCES

[1] Alciatore, D. (1989) *Path-Planning Algorithms for Pipe Manipulator*, Ph.D. Dissertation, University of Texas at Austin.

[2] Davidor, Y. (1991) *Genetic Algorithms and Robotics: A Heuristic Study for Optimization,* World Scientific Publishing Company.

[3] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing company.

[4] Hsieh, Ting-Ya and Hass Carl, T. (1993) *Applications of Large Scale Manipulators in the Construction Environment*, Proceedings of the 10th International Symposium on Automation and Robotics in Construction, Houston, Texas, USA, 55-62.

[5] Latombe, J.C. (1991) *Robot Motion Planning*, Kluwer Academic Publishers, Norwell, Massachusetts.

[6] Lozano-Perez, T. (1983) *Spatial Planning: A Configuration Space Approach*, IEEE Transactions on Computers, C-32(2), 108-120.

[7] Mark AC Gill and Albert Zomaya (1998) *Obstacle Avoidance in Multirobot Systems: Experiments in Parallel Genetic Algorithms,* World Scientific Publishing Company.

[8] Morad, A. and Beliveau, J. (1992) *Path Finder-AI based Path Planning System*, ASCE Journal of Computing in Civil Engineering, 6, 114-128.

[9] Paredis, J. (1996) C*o-evolutionary Computation,* Artificial Life Journal, 2(4), 355-375, MIT Press.

[10] Raghunatha Reddy, H. (1997) *Automated Path Planning for Crane Lifts*, M.S. Thesis, Department of Civil Engineering, Indian Institute of Technology Madras.

[11] Schwartz, J.T. and Sharir, M. (1988) *A Survey of Motion Planning and Related Geometric Algorithms*, Artificial Intelligence, 37, 157-169.

[12] Udupa, S. (1977) *Collision Detection and Avoidance in Computer Controlled Manipulators*, Ph.D. Dissertation, Department of Electrical Engineering, California Institute of Technology, California.

[13] Sivakumar, PL., Varghese, K. and Ramesh Babu, N. (1999) *Path Planning of Construction Manipulators using Genetic Algorithms*, Proceedings of the 16th International Symposium on Automation and Robotics in Construction, Madrid, Spain, 555-560.