DEVELOPMENT OF A CONSTRUCTION WORK CELL FOR COOPERATIVE LIFT PLANNING

M.S. Ajmal Deen Ali, S. Vijaya Bhaskar, N. Ramesh Babu and Koshy Varghese*

Dept. of Mechanical Eng., I.I.T. Madras, India, nrbabu@iitm.ac.in
**Dept. of Civil Eng., I.I.T Madras, India, koshy@iitm.ac.in*

ABSTRACT: Cooperative crane lifts are considered as an alternative to specialized heavy crane lifts. However, in executing a cooperative crane lift, the precise coordination among the cranes is critical. This level of precision is better assured through automation of the lift operation. However, due to the drawbacks in current crane technology automated execution is not feasible. The development of a prototype work cell for cooperative lifts is addressed in this paper. The prototype manipulator developed has four degrees of freedom and possess the degrees of freedom of a typical hydraulic crane -swinging, luffing, telescoping and hoisting. For programmable control, the manipulator is interfaced to the computer. Uncoordinated motion among the individual manipulators often occurs when subject to cooperative manipulation. The cooperative manipulation is ensured by means of a typical logic that paves the way for proper control flow sequence between the individual controllers through RS-232 serial port. A series of trials were conducted to investigate the behaviour of the system under different cases.

KEYWORDS: Crane lift, Cooperative manipulator, Automated path planning, Computer controlled execution.

1. INTRODUCTION

Safety, productivity and reliability are the three predominant issues facing the construction industry today. Construction work in general is very labor intensive and conducted in situations that are dangerous. It is due to this reason that construction robotics has been a very active research area in the construction industry. Among them, building robots have been employed in various tasks, including material handling and various interior and exterior finishing works. A Local Area Network (LAN) based building maintenance and surveillance robot [1] for maintenance work and for building safety was developed. Applications of robots in hazardous working environment like shotcreting [2], removing lead based paints [3] were developed to replace human operators from harmful effects.

A kinematic control of a pneumatic system by hybrid fuzzy PID was developed [4]. This control algorithm has the capability to position the piston at any desired point in the entire stroke length of the cylinder. Hydraulic large range robots for construction application many a times result in extreme non-linearity. Suitable control methods based on model based dynamic response of the system for position controls are discussed [5]. An automated path planning of cooperative crane lifts using co-evolutionary genetic algorithm was discussed [6]. The issues were focused at developing an automated path-planning procedure for cooperative lift planning using genetic search procedures. The manual operator did the planned lift execution.

The risk associated with cooperative lift execution is much higher than that of a single crane lift due to the interaction among cooperative cranes [7]. In summary, the risks associated with cooperative crane lifts can be categorized into (1) Risk of human error in preparing lift plans and (2) Risk of uncoordinated motion during the lift plan execution. These risks can be overcome by (i) Automating the planning tasks using computer aided planning tools and (ii) Automating lift plan execution using suitable control systems to ensure coordination among cranes.

This paper address the second issue i.e. risk of uncoordinated motion during the lift plan execution. The risk factor associated with uncoordinated motion can be eliminated by means retrofitting the existing cranes with technology for programmable control and automating the lift execution using suitable control systems. Towards this objective, a prototype cooperative manipulator system is developed. The prototype manipulator is an electrically controlled arm model for pick and place operation. The cooperative manipulator system can be programmed to follow the desired trajectory for collision free movement of the object from pick to place location.

2. MOTION PLANNING SEQUENCE

Fig. 2 illustrates the components of a program that automatically generates collision free motions for the cooperative manipulator system. Collision free motion planning relies on three dimensional (3D) information about the physical structure of the robot as well as the workspace in which the robot has to operate. So, the first step in collision free motion planning is the 3D *world description acquisition*. This contains information about the kinematic structure of the manipulator system, obstacles in the work environment and their 3D information such as position and orientation along with details of the lift object.

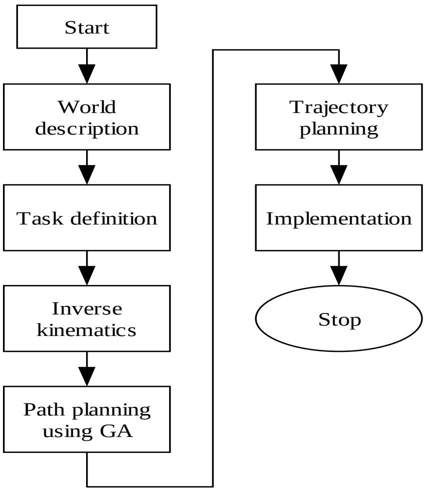


Fig. 2. Program flow control

The *task definition phase* defines the overall task of the manipulator system i.e. defining the desired Cartesian pick point and place point. The *inverse manipulator kinematic* deal with the computation of the set of joint angles and linear translations for the arms that will set the hook point or the end effector at the defined Cartesian pick point or place point. The path planner uses this information and employs a search technique to find a collision free path from the start configuration of the manipulator system to its goal configuration. The trajectory planner converts the collision free path into a trajectory that can be executed by the manipulator i.e. the trajectory planner produces motion commands for the servomechanism of the manipulator. These commands are executed during the implementation phase.

3. PATH PLANNING

A collision free path planner essentially consists of two important components: a search algorithm and collision detection algorithm. The search algorithm explores the search space for optimum movement of the manipulator system from pick to place location. As the search progresses, the collision detection algorithm checks the feasibility of each step movement i.e. intermediate configuration's feasibility in the search space. It checks for collision of the manipulator system with other structural components in the workspace of the robot. A schematic figure representing the workspace along with the manipulator system is shown in Fig. 3a. **T**he planned path is shown in Fig. 3b.

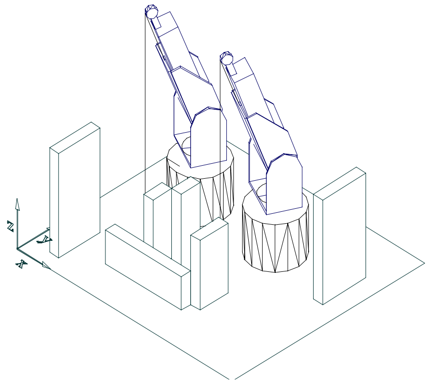


Fig. 3a. Schematic of the workspace

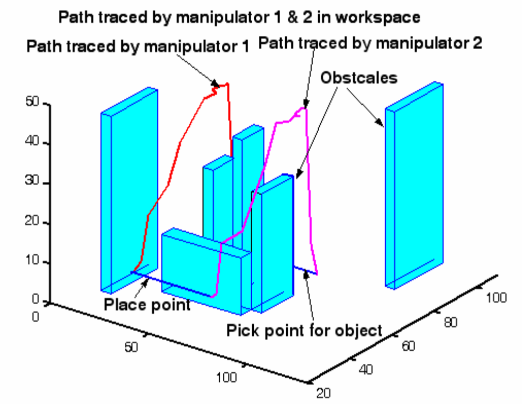


Fig. 3b. Schematic of the planned path

4. MOTION PLANNING

In Fig. 4a, a functional model of a manipulator system is shown. It is a 4 DOF manipulator with two revolute joints and two prismatic joints. The manipulator's specifications are given in Table 1 and the schematic diagram is shown in Fig. 4b. The main column along with other links can rotate around a fixed base on roller bearings. According to crane terminology, it is referred as base and its DOF is named as swinging. The second link is named as boom and its DOF is referred as luffing. The third link is called as extended boom and its DOF is referred as telescoping. The fourth link is the flexible rope that usually gets wound on a drum placed at the end of the third link. The drum is rotated by servomotor. The link is referred as hoist rope and DOF is called as hoisting.

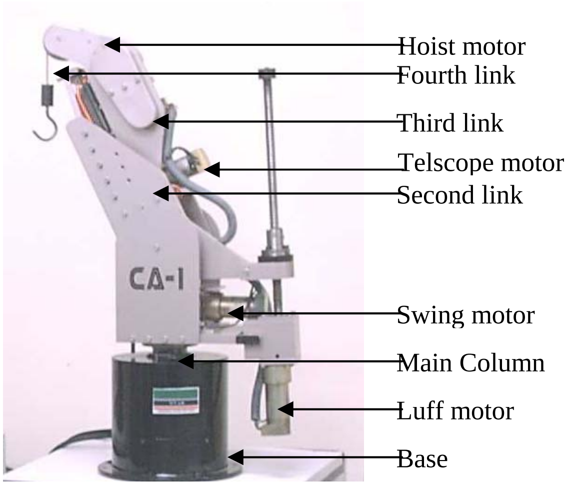


Fig. 4a. Functional model of a manipulator System

Configuration	Revolute type
Axes of freedom	4
Pay Load	10 kg
Base-Swinging	360^0
Boom-Luffing	80^0
Telescoping	150 mm
Hoisting Height	Depends on Luffing Angle of Boom, Telescoping and geometry of Load

Table 1. Manipulator's specification

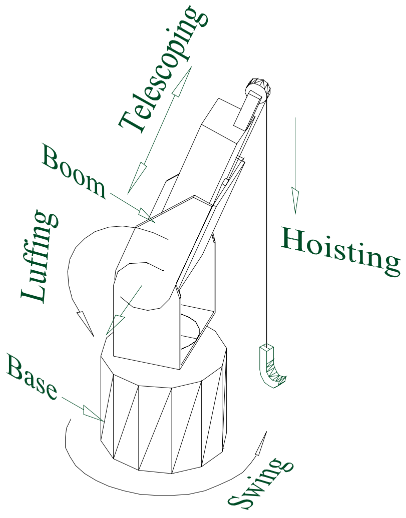


Fig. 4b. Schematic of the manipulator system

For attaching the load, the hoist rope end is provided with a hook arrangement. The manipulator system has repeatability of ±0.3mm for all the links. The manipulator has hardware stop position also called as home position and acts as the manipulator's reference point of move for individual axes.

5. KINEMATIC CONTROL

A block diagram of the overall system is shown in Fig. 5a. To accomplish a kinematic control of a manipulator system, the following considerations are required:

1. Designing an interface unit to convert the supplied data to a form that can be interpreted by the control unit
2. Developing a control algorithm and programming the control unit for position control of the manipulator links
3. Drive unit design in order to amplify output signal from the controller unit to drive the manipulator links.
4. Position sensing unit in the form of encoder; sense the actual positional move and send information to the control unit for proper corrective action.

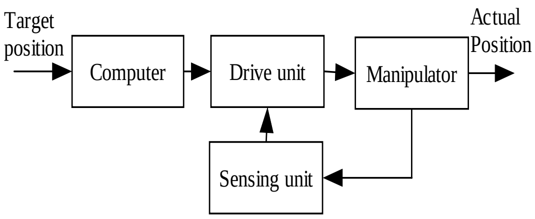


Fig. 5. Block diagram of the overall kinematic control

6. TASK SEQUENCE IN MOTION CONTROL

Fig. 6a represents the controller connection. In Fig. 6b, a flow chart representing the motion control sequence is shown. The manipulator is controlled from a host computer through a host processor, which in turn connects to a programmable multi-axes controller (PMAC) card. The PMAC is interfaced to the servo-drives, which in turn activates the motor. The motor actuation results in link displacement. The encoder sends the actual position feed back to the motion chip where corrective action will be taken.

The manipulator *initializes* the links by making all of them to return to their home position. The *user interface* dialog box written in C++ resides in the host computer. The user enters the desired link position data into the interface unit. The motor used is a servomotor, therefore the position data is then converted to the required pulse width modulated (PWM) counts for driving the motor.

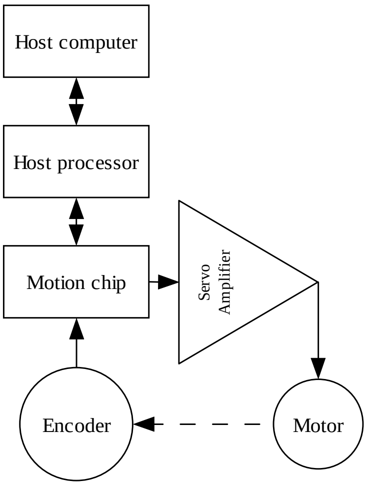


Fig. 6a. Block diagram for controller Connection

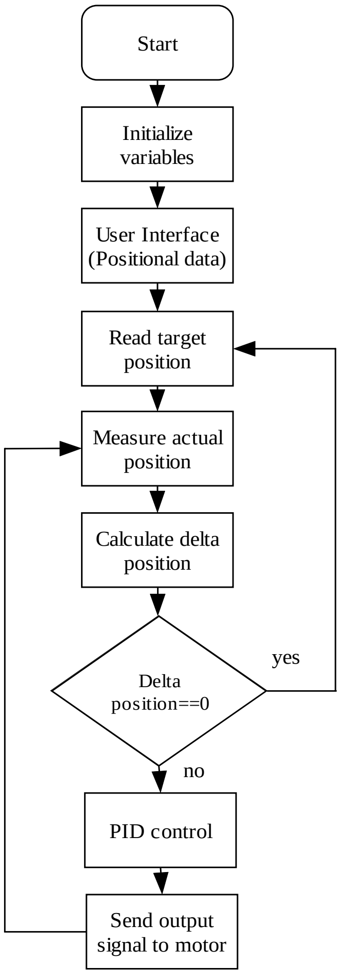


Fig. 6b. A flow chart representing the motion control sequence

The PWM counts required for one complete revolution of respective links varies depend on the servomotor used and the kinematic structure of the link. The equation for calculating the PWM counts is given in the equation 6a:

$$\left(PWC_c\right)=\left(N_2\right)\left(N_1\right)\left(CPR\right)\left(T_2/T_1\right) \qquad (6a)$$

where:
PWC_c is PWM counts required for desired rotation of driven gear (N_2), N_1 is number of revolutions of the driver gear, N_2 is number of revolutions of the driven gear, CPR is encoder counts per revolution of the driver gear, T_1 is number of tooth on the driver gear and T_2 is number of tooth on the driven gear.

Actual position is measured in the form of digital signal, by the *encoder*, which is present as an integral part of the motor shaft. The *control unit* then compares the actual position with the desired position and determines position difference. This difference has to adjust towards zero. A *proportional-plus-integral-plus-derivative (PID)* control algorithm is used to control the motor, responsible for desired link displacement. The PID control algorithm provides a full digital lead compensation for closed loop system stability and the motor command is the output. The compensation D(z) has the form as shown in the *equation 6b*:

$$D(z)=\frac{k\left(z-\frac{A}{256}\right)}{4\left(z+\frac{B}{256}\right)} \qquad (6b)$$

where:
z is digital domain operator, K is digital filter gain, A is digital filter zero and B is digital filter pole.

The compensation is first order lead filter which in combination with the sample timer T affects the dynamic step response and stability of the control system. The sample timer, T, determines the rate at which the control algorithm gets executed. All parameters A, B, K and T are 8 bit scalars.

The digital filter uses previously sampled data to calculate D(z) as shown in *equation 6c.*

$$MC_n=\left(\frac{K}{4}\right)\left(X_n\right)-\left[\left(\frac{A}{256}\right)\left(\frac{K}{4}\right)\left(X_{n-1}\right)+\left(MC_{n-1}\right)\left(\frac{B}{256}\right)\right] \qquad (6c)$$

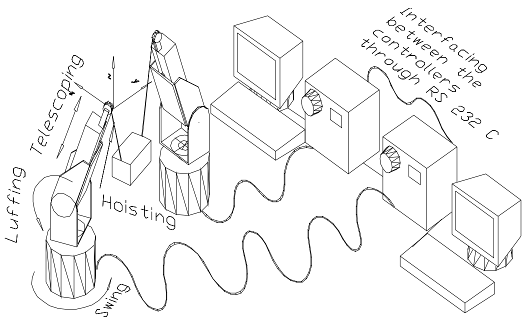
where:
n is current sample time, n-1 is previous sample time, MC_n is motor command output at n, MC_{n-1} is motor command output at n-1, X_n is (command position-

actual position) at n and X_{n-1} is (command position-actual position) at n-1

The motor command output is then sent to the PWM port. The PWM port outputs the motor command as a pulse width modulated signal with the correct sign of polarity. The control process is repeated until the link reaches the acceptable target position.

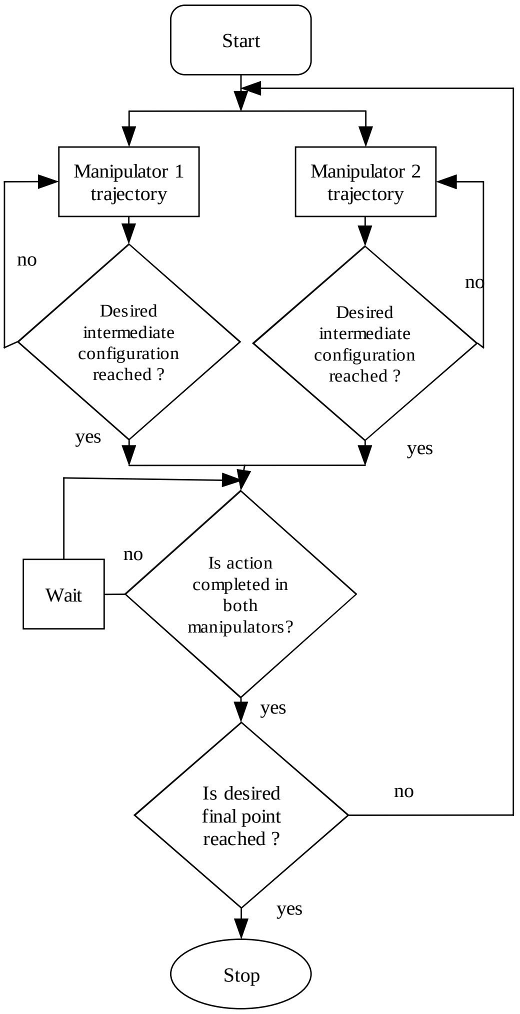
7. COOPERATIVE MANIPULATOR CONTROL

The schematic diagram of the cooperative manipulator system is shown in Fig. 7a and a flow chart representing the cooperative control is shown in Fig. 7b. Cooperative manipulation of the object from pick to place position is ensured only when manipulator 1 and manipulator 2 follow their prescribed trajectories in proper time domain. These trajectories have same number of intermediate configurations between pick and place location.



7a. Schematic diagram of the cooperative manipulator system

Although both manipulator systems are identical, they are found to have different mechanical and electrical characteristics. This results in execution time delay between the manipulators when they are operated independently to follow their prescribed trajectories. For example, manipulator 1 may be at its second intermediate point in its trajectory whereas manipulator 2 could have traveled to the third intermediate point in its trajectory. This leads to uncoordinated motion between the manipulators. This has necessitated the need to control these manipulators for cooperative manipulation. This is possible through RS232 serial port communication between the controllers i.e. controller 1 and controller 2 communicate between themselves through RS 232. Communication between the controllers results in knowing the status of first controller by the second and vice-versa.
A typical logic, constructed inside the controllers initiates the new task only when the previous task assigned to the controllers were executed i.e. both manipulators had reached their same intermediate point.



7b. Flow chart representing cooperative control

This aids in proper coordination among the manipulators for following their prescribed trajectories in proper time domain for achieving cooperative manipulation.

8. RESULTS

Once the collision free path is planned, the planned path is converted to the respective trajectory profile. Using a PID control algorithm, appropriate motor command outputs are generated with respect to time domain and sent to the motor for proper motion execution.

The cooperative manipulation between the manipulators is ensured by means of a typical logic that paves the way for proper control flow sequence among the independent controllers through RS-232 serial port. This results in coordinated motion between the independent manipulators. Snapshots of the collision free motion execution by the manipulator system through the workspace are presented in **Figs. 8(a)-(d).**

The cooperative manipulator system picks up the object, hoist the object over the obstacles, moves progressively over the constrained space, hoist down and place the object on the desired location.

Fig. 8a. Pick location

Fig. 8b. 5th Intermediate location

Fig. 8c. 14thIntermediate location

Fig. 8d. Place location

9. CONCLUSION

The objectives of this study were achieved by designing a crane manipulator and controlling it through computer control, for flexible heavy lift applications. The movement of the manipulator system was controlled to move to any desired point within its workspace boundary. All the associated function with heavy lift operation like planning and execution is completely automated. The user has to enter the pick point and place point for the cooperative manipulator system in the user interface module. The executable search algorithm starts generating the collision free path. The collision free path configurations are then converted to the desired trajectories and fed to the respective control unit for desired moves in the workspace environment. The full scale development of this protype heavy lift system can reduce the risks associated with manual execution of cooperative lifts. .

Acknowledgements

The authors wish to acknowledge the Department of Science and Technology (DST), New Delhi, India, for the financial support given to this research work vide Grant No. III. 5(23)/2000-ET.

REFERENCES

[1] So, T. P. Albert., Chan, W.L., “Lan-based building maintenance and surveillance robot.” *Automation in Construction*, 11, 619-627, 2002.

[2] Cheng, Min-Yuan., Liang, Yueh., Wey, Ching-Ming., Chen, Jiann-Chyun., “ Technological enhancement and creation of a computer aided construction system for the shotcreting robot.” *Automation in Construction*, 10, 517-526, 2001.

[3] Lorenc, Stevan J., Handlon, Brian E., Bernold, Leonhard E., “ Development of a robotic bridge maintenance system.” *Automation in Construction*, 9, 251-258, 2000.

[4] Parnichkun, Manukid., Ngaecharoenkul, Charoen., “Kinematics control of a pneumatic system by hybrid fuzzy PID.” *Mechatronics*, 11, 1001-1023, 2001

[5] Kuntze, H. B., Hirsch, U., Jacubasch, A., Eberle. F., Goller, B., “ On the dynamic control of a hydraulic large range robot for construction application.” *Automation in Construction*, 4, 61-73, 1995.

[6] Ali, M. S. Ajmal Deen., Babu, N. Ramesh., Varghese, Koshy., “Offline path planning of cooperative manipulators using co-evolutionary genetic algorithm.” *19th International Symposium on Automation and Robotics in Construction*, 415-424, 2002.

[7] Shapiro, H. I., Shapiro, J. P., and Shapiro, L. K., *Cranes and Derricks*, 2nd Edition, McGraw-Hill, Inc, New York, N.Y, 1991.