Flexible Control of Spatial Hyper Reundant Robot

Arun J.[1], SundarGanesh CS[2], Arunmozhi R[3], Parthiban M[4], Ponnusamy R[5], Lakshmipathy P[6]

[1, 3,4,5,6] UG Student, Department of EEE, PSG College of Technology, Coimbatore, India

[2]Assistant Professor, Department of Robotics & Automation, PSG College of Technology, Coimbatore, India

Abstract: Industrial robots manipulate their end effector in a pre-defined work-space. Structures are made rigid to cater the pay load requirements. Navigation of these arm inspired robots in cluttered environment poses a challenging task. This precisely defines the requirement of a redundant manipulator wherein the controllable degrees of freedom are more than the total degrees of freedom in the work space. Present work is focused on design of one such manipulator that can function in a complex environment. This serial manipulator named as Multilink Spatial Hyper Redundant Robot comprises of sixteen links and fifteen joints. With each joint providing a prescribed degree of freedom a thirty degrees of freedom is realized on the system. Manipulator's parts were manufactured using selective laser sintering process, a rapid prototyping technique. Degrees of freedom in the system were precisely controlled using ropes. Ropes were driven by rotary stepper motor that serves as a drive for the manipulator. Links were provided with provision for anchorage of ropes. Dummy link was introduced in the assembly for increasing the work volume of the robot. Springs provided in the assembly serve as retention members. Stepper motors were used to actuate the ropes, as it facili-tates to control the robot with an open loop control system. The PCB required for the drivers were designed using ARES 7 professional software, fabricated and assembled.

Keywords: Spatial hyper redundant robot, Stepper motor driver, flexible link control, Microcontroller

I. Introduction

The slave arm robot, based on human arm has rudimental control and little perception about the environment. Structures of these robots are designed as rigid modules considering the payload. Advent of technologies replacing man power with machines and demand for human safety in work environment sparked the study of field robotics. These robots necessitate a high degree of flexibility to maneuver in hazardous work environment such as inspection and repair in nuclear power plants, which is not met by the conventional robots. Traditional approach of developing structures that would simulate a human action needs to be considered for this purpose. Several researches pertaining to the locomotive mechanisms (reptiles), prehensile nature (Mon-keys tail), grasping ability (elephant trunk) and movement of invertebrate organs (Octopus legs) were successfully studied and these ideas translated into robotic application. For an example, a rope drive controlled serpentine robot used in rescue operations that navigate with gait motion has been developed previously [1-3].

In 1940, the ethics of interaction between robot and human was envisioned to be governed by three fundamental laws of Isaac Asimov, Russian science fiction writer in his novel. Middle of twentieth century was the first exploration of artificial intelligence. This evolved robot-ics which refers to science and technology of robots. The first slave arm robot which duplicated the human arm had rudimental control and little perception about the environment. Late twentieth century saw development in integrated circuit, digital computers etc. that facilitated computer controlled robots to be designed and programmed known as Industrial robots. In 1980, robotics was defined as the science that studies the intelligent connection between perception and action. The era which lit the present project was during 1990 when research was boosted by the need to rely on robots to address human safety in hazardous environment (*field robotics*), or to enhance the human operator ability and reduce fatigue (*human augmentation),* or else by the desire to develop product with potential market aimed at improving quality of life *(service robotics).* By the dawn of millennium, robotics has undergone a major transformation in scope and dimensions.

Robots are made of links and joints. Joints deliver links with required degrees of freedom. Links and joints are manipulated together inside a work volume. This refers to a space within which the robot can manipulate its wrist end. Commercial robots are specified based on its payload capability handle at maximum operating positions. The manipulator itself can be of two types namely, Serial manipulator and Parallel manipulator .Serial manipulators, otherwise known as open loop manipulators consist of one fixed end and other a free end. Series of links and joints manipulate to position the end effec-tor inside the work space. A parallel manipulator is a closed loop kinematic chain mechanism whose end effector is linked to the base by several independent kinematic chains. Several configurations can be realized in the field of robotics. Most common types are Polar configuration, Cylindrical configuration, Cartesian configuration and Jointed arm configuration.

II. LITERATURE SURVEY

Spatial Hyper Redundant (SHR) robot has more controllable DOF than actual DOF due to hyper redundancy. stepper motors are used to actuate the links. The control system for an 8 link snake like robot using CAN protocol. A node consists of a microcontroller, CAN controller, CAN transceiver and a physical medium for transmission [1].

A novel spatial hyper redundant manipulator inspired in the motions of the worms is introduced. The displacement analysis is presented in a semi -closed form solution, whereas the velocity and acceleration are carried out by means of the theory of screws [2].

A scheme is developed that evaluates the score of trial kinematic configurations while accounting for various geometric constraints. This versatile method is based on a simultaneous search for bot h the best curve capturing the overall kinematic robot posture [3].

A kinematic model has framed for a spatial hyper redundant manipulator and verified the experimental results with results obtained from kinematic analysis. They used numerical approach for the kinematic study rather than using a analytical approach. [4].A case study had done on how the snake robots are be-ing implemented to repair pipes in a nuclear power plant. First a model of the nuclear reactor was built to do mock tests and the robots performed up to the acceptance levels. The whole process took only three days using the snake arm robots. [5].The flexible arm is fabricated in which has individual controllable segments connected in series. The purpose of this flexible arm is to provide good rigidity in the bending plane of the element and a high torsion resistance. This has been achieved by each element being designed with arched opposed single-or double-curved segments which with the curved surfaces of each segment engaging the surfaces of the adjacent segments, whereby elements are arranged when actuated.[6] .The link assembly for a robot arm which comprises of first and second link each adapted for a limited movement. These links are keyed or bonded through a resilient elastomeric material.[7].The algorithm is discussed for controlling hyper-redundant manipulators by using three strings.[8].A modal approach in which a set of intrinsic backbone curve shape functions are limited to a modal form[9]. A hyper redundant robot elephant trunk type with truncated cone structure. Flexible structure is made of two independent modules that drive, bending robot is made by nature. Drive block for each module consists of two stepper motors that al low bending toward the axes X and Y [10].

III. ELECTRICAL DESIGN

The requirements of electrical drive system for a continuum robot are high holding torque to maintain the robot in a specified shape, positional accuracy and repeatability. Servo motors, linear stepper motors and hybrid stepper motors easily satisfied all the requirements. Since our design is based on open loop system, stepper motor drive system is used. It is cost effective than servo and linear stepper motors

Selection of Motor

A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotation. Every revolution of the stepper motor is divided into a discrete number of steps, and the motor must be sent a separate pulse for each step. The stepper motor can only take one step at a time and each step is the same size. Since each pulse causes the motor to rotate a precise angle, typically 1.8°, the motor's position can be controlled without any feedback mechanism. As the digital pulses increase in frequency, the step movement changes into continuous rotation, with the speed of rotation directly proportional to the frequency of the pulses. Hybrid stepper motors are named because they use a combination of permanent magnet and variable reluctance techniques to achieve maximum power in a small package size. Stepper motor "step modes" include Full, Half and Microstep. The type of step mode output of any stepper motor is dependent on the design of the driver.

FULL-STEP Standard hybrid stepping motors have 200 rotor teeth, or 200 full steps per revolution of the motor shaft. Dividing the 200 steps into the 360º of rotation equals a 1.8º full step angle. Normally, full step mode is achieved by energizing both windings while reversing the current alternately. Essentially one digital pulse from the driver is equivalent to one step.

HALF-STEP Half step simply means that the step motor is rotating at 400 steps per revolution. In this mode, one winding is energized and then two windings are energized alternately, caus-ing the rotor to rotate at half the distance, or 0.9º. Although it provides approximately 30% less torque, half-step mode produces a smoother motion than full-step mode.

MICROSTEP Micro stepping is a relatively new stepper motor technology that controls the current in the motor winding to a degree that further subdivides the number of positions between poles. Micro stepping drives are capable of dividing a full step (1.8º) into 256 micro steps, resulting in 51,200 steps per revolution (0.007º/step). Micro stepping is typically used in applications that require accurate positioning and smoother motion over a wide range of speeds. Like the half-step mode, micro stepping provides approximately 30% less torque than full-step mode.

The motor we have selected is hybrid bipolar stepper motor Motor specification.

A Stepping Motor System consists of three basic elements, often combined with some type of user interface. The Indexer (or Controller) is an IC capable of generating step pulses and direction signals for the driver according to given input parameters i) Number of steps ii) Direction iii) Stepping mode

Table 1 Stepper Motor Specification

S. No.	Parameter	Value
1	Voltage	2.3 V
2	Current per phase	1.5
3	Holding torque	4 kg-cm
4	Degree per step	1.8°

The Driver (or Amplifier) converts the indexer command signals into the power necessary to energize the motor windings.

The circuit was designed as a PCB in ISIS ARES, fabricated and assembled.The PCB Design of driver circuit is shown in figure 1.

Our stepper motor driver is based on L297 and L298.L297 acts as the indexer and L298 as the amplifier. The Fabricated PCB driver board is shown in fig.2.The L297 integrates all the control circuitry required to control bipolar and unipolar stepper motors. When used with a dual bridgedriver such as the L298N forms a complete microprocessor-to-bipolar stepper motor interface.

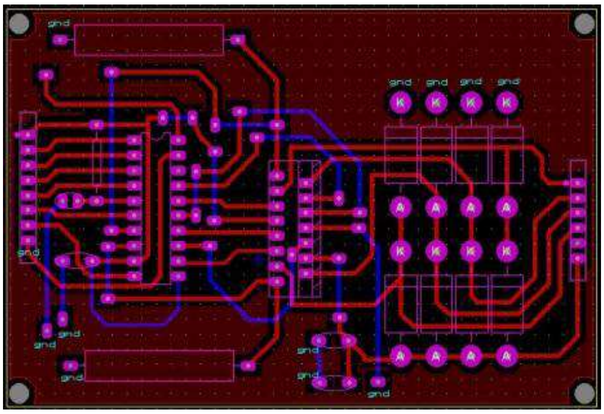

Fig.1 PCB Design of Driver Circuit

Fig.2 Fabricated PCB driver board for a single Stepper Motor

IV. CONTROL STRATEGY

The prototype robot used here has sixteen links in which four are targeted links. The rest are dummy links to make the body of the robot. Each targeted link has four strings connected to it whose lengths control the position and orientation of the robot. The strings are connected to stepper motors through pulleys. Algorithm to control the robot is executed in MATLAB. MATLAB program provides the length each string has to be moved in positive and negative direction, to reach the desired orientation. These values of length are given to the microcontroller which generates the control signals and sends it to the respective stepper motor drivers. The drivers drive the stepper motors for the required length and direction. The block diagram for controlling the entire robot is shown in fig.3.

Since the control algorithm is separated from generation of control signals, the algorithm can be modified as required without affecting the controller program as long as it is in a specified format. For example the algorithm that is being used here is a simple algorithm to move the end effector of the robot to a specified position in space without taking the shape and orientation of the robot into consideration.

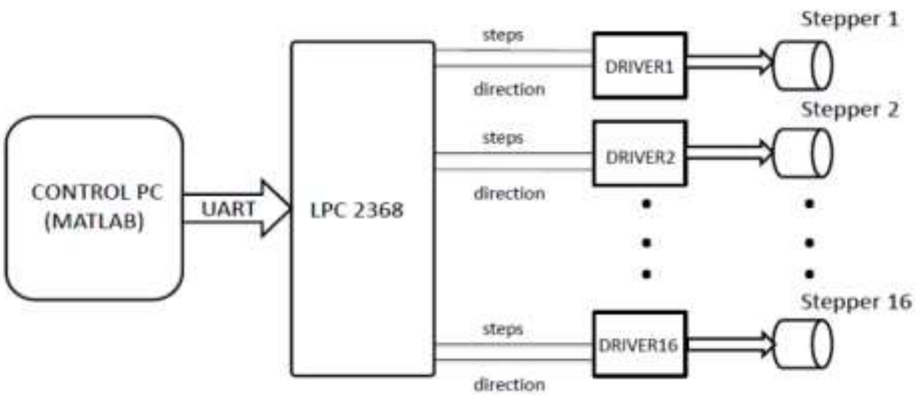

Fig.3 Block diagram for Controlling the Entire Robot Setup

The algorithm to be coded in MATLAB for controlling the robot is based on three string contol.It has been modified so as to suit to control of hyper redundant manipulator by four strings instead of three strings. Input parameters of each targeted link are θ and α. θ and α of all the four targeted links are entered to the MATLAB program. It is shown in fig.4.

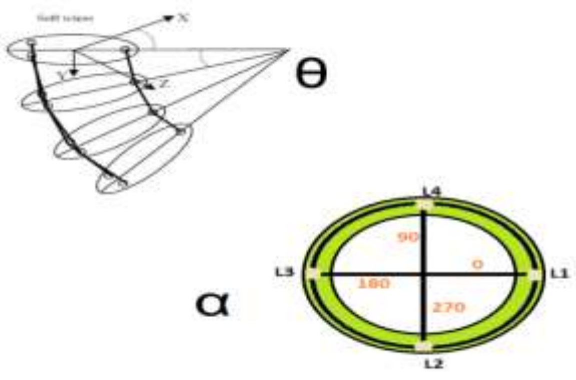

Fig.4 Input Parameters α and θ for each Target Link

The program calculates the lengths to which each strings is to be moved. The program further calculates the direction and the number of steps (step count) each stepper motor should move. These data are sent to the microcon-troller via UART communication system established between MATLAB and microcontroller.

Programming of LPC

Programming was done on MCB2300 which is the development board for LPC2368 which was used to provide control signals to the stepper motor drivers. The programming and simulation was done using Keil µVision 4 IDE. For smooth operation of the robot it is required that all the motors start and stop functioning at the same time. So the primary function of the microcontroller is to calculate the speeds of each motor and generate signals accordingly so that the above mentioned criterion is satisfied.

The input of the microcontroller is data sent from MATLAB. The data is number of steps and direction of each stepper motor. The microcontroller receives this data, calculates the speed of each motor and generates control signals.

The microcontroller generates two control signals per motor making it 32 signals in total. One is a square signal. Number of rising edges of this signal gives the number of steps. The frequency of this signal gives the speed of the motor in steps/second. Other signal takes high or low states and decides the direction of rotation.

The program was written in C language using the Keil µVision 4 IDE. The micro-controller is required to generate sixteen independent square signals. All the sixteen signals may have different frequencies and different number of rising edges. Also sixteen direction signals were generated. A time period of five seconds was assumed for the robot to move from one position to another. This makes the frequency of square signal directly proportional to the number of steps. After receiving the number of steps from MATLAB, the microcontroller calculates all the frequencies and the corresponding time periods. LPC2368 has four timers with each having four match registers. Half the value of calculated time periods is loaded in the sixteen match registers. All the match registers can generate interrupt upon match. When the timer value matches the value in one of the match registers, the interrupt service routine handles two functions. Corresponding output pin is toggled and value in the match register is increased by its initial value. Subsequent operation generates a square signal. Same pattern of operation is followed for all the sixteen square signals. The direction pin goes high for clockwise rotation and goes low for counter clockwise rotation of stepper motor.

Algorithm in MATLAB calculates the number of steps and direction of each step-per motor. This data is sent to LPC2368 via serial communication. UART 0 of LPC2368 was used for receiving data from MATLAB. Communication settings were coded in MATLAB, and UART 0 of LPC2368 was also configured with the same settings. An USB-RS232 cable was used to connect the computer with MATLAB and LPC 2368.

V. RESULT

ARM7 microcontroller of NXP was used to generate the control signals given to the driver. Control signal from the microcontroller controls the number of steps, speed and the direction of all the stepper motors. Signals of required number of steps and frequency were produced by the microcontroller. Algorithm to control the robots position and orientation was developed for controlling each targeted link using four ropes. It was coded in MATLAB. When the input parameters are entered to MATLAB it computes the change in length of every rope. This value is given to the microcontroller which generates the required signals to bring about the change in length of ropes which consequently controls the position of the robot. UART communication was used to transmit data from MATLAB to microcontroller. ISIS 7 PROFESSIONAL software was used to simulate the system. The complete control of the robot was executed in a step by step manner.

CONTROL OF FOUR LINKS

Initially only four stepper motor, four drivers, algorithm required to control only four links and corresponding microcontroller program was developed and tested. The targeted link was successfully moved to any achievable position using our algorithm. The arrangement of four link is shown in fig.5

Fig.5 Control of Four Links

CONTROL OF EIGHT LINKS

After successful control of one complete set, another set of four links were added and algorithm for eight links was tested. Initially there were problems with motor current consumption and the motors started to run randomly without any control signals form controller. When all eight motors were given supply simultaneously the total current consumption was lesser than the sum of currents the motor consumed when it was supplied individually. The microcontroller power supply was given from a separate power source instead of being powered from laptop and all the output pins of the controller were made high. But the problem still persisted and motors ran randomly. Then opto-couplers were used to isolate the control signals from controller to the power circuit. With this the problem was immensely reduced but not completely eliminated as the random starting still happened only on rare occasions. Thus two sets with eight links were tested using the algorithm for eight links and found that there was slight error in the final position actually achieved by the robot. The arrangement of eight link is shown in fig.6Hence the algorithm was suitably modified to compensate for this error and eight links were controlled successfully.

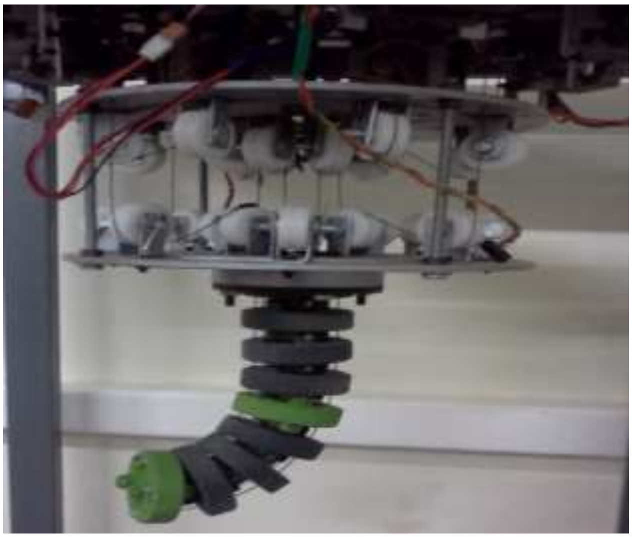

Fig.6 Control of Eight Links

CONTROL OF SIXTEEN LINKS

The complete robot was assembled and the drivers for all sixteen motors were fabricated and assembled. Since a single linear power supply was not enough to supply all sixteen motors, a 200W SMPS was used to power the motors. But the random starting of motors occurred again in spite of using optocouplers. Hence two separate power sources in parallel were used to supply the motors and the entire setup was controlled. The arrangement of sixteen link is shown in fig.7

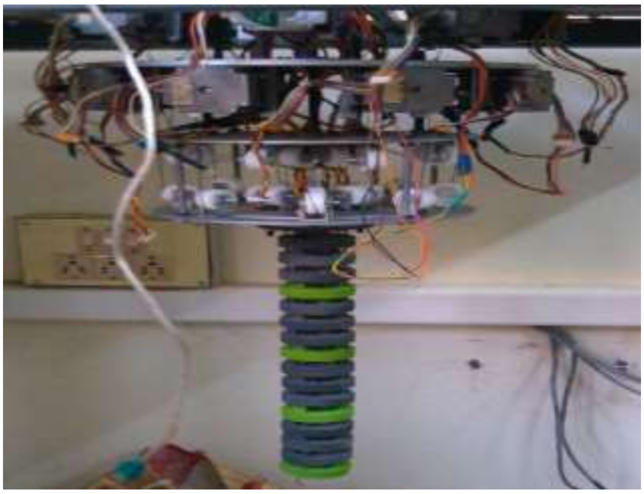

Fig.7 Control of sixteen Links

VI. CONCLUSION AND FUTURE WORK

A heuristic approach followed for design of manipulator demands development of more than one prototype. The final design of manipulator is an outcome of series of design optimization made on several such prototypes. Rapid prototyping technique was used for prototyping sixteen links and fifteen joints. Selection of stepper motor was substantiated by torque calculation. Both actuator pack mountings and drive pulley mountings were fabricated. Stepper motor drivers were designed and manufactured. The algorithm was coded in MATLAB. On assemblage of all the above components a working prototype was realized. The developed model is provided with thirty two degrees of freedom to the entirety of the system. This could be further increased by addition of dummy links. Present work would serve as a viable source for future studies pertaining to redundancy in robots.

MATLAB offers only a single set of inverse solution which holds good for testing purposes. The very objective of the robot is navigation in a complex environment, hence, an intelligent obstacle avoiding solution is required, which demands selection of best possible solution from a set of inverse solutions. A better user interface with facilities like GUI can be developed to control the robot. The present prototype implements an open loop control. Feedback devices can be attached and a closed loop control system with greater accuracy can be developed.

VII. References

[1] N. Kiranmayee, R. Seethalakshmi , Jobby C. Johnson, Joel Jose, S. Sakthivel and S.A. Sujitha (2002), A Distributed Control System for Spatial Hyper Redundant Robot using CAN, Journal of Artificial Intelligence, Vol5, pp.152-160.
[2] Jaime Gallardo-avalardo, Raul-lesso-arrayo.J. Santos Garcia Miranda (2011), A worm inspired new spatial hyper redundant Manipulator, Journal of Robotica,Vol. 29, pp.571-579
[3] Abdel hakim Chibani, Chawki Mahfoudi,Taha Chettibi, Rochdi Merzouki, Abdelouhab Zaatri (2015),Generating optimal reference kinematic configurations for hyper-redundant parallel robots, Proceedings of the Institution of Mechanical Engineers, Part I:Journal of Systems and Control Engineering.
[4] Hannan M.W and Walker I.D, "Novel Kinematics for Continuum Robots (1993), IEEE International Conference on Robotic and Automation. Springer, pp. 227-238.
[5] Rob Buckingham (2005), Snaking Around in a Nuclear Jungle, Industrial Robot: An International Robot, Vol. 32, pp 120 – 127.
[6] Larson (1983), United States Patent, App No.351, 589, Filed Feb, 23, 1982.
[7] Buckingham (2009), United States Patent, App No: 10736, 354, Filed Dec, 15, 2008
[8] Ivanescu Mircea and Cojocaru Dorian(1994), Hyper Redundant Manipulators, University of Craiova, Romania, IEEE Robotics Magazine, pp. 22-29
[9] Gregory S,Chirikjian and Joel W Burdick(1994),A Modal approach to Hyper Redundant Manipulator Dynamics, IEEE Transactions on Robotics and Automation.Vol-10, pp.343-354.
[10]Vladu, Ileana; Strimbeanu, Daniel; Ivanescu, Mircea; Bizdoaca, Nicu; Vladu, Ionel Cristian; Florescu, Mihaela Cecilia (2011), Control System for a Hyper-Redundant Robot, Information Control Problems in Manufacturing, Vol. 14, pp. 853-858.

