Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 5 (2017) pp. 011-019

Simplified Architecture of a Hyper Redundant Robot for Wheel-Less Locomotion

Ginu Paul A, Ritwik Aravind

Department of Mechanical Engineering, Toc H Institute of Science and Technology, Arakkunnam, India-682313

ginupaul@gmail.com, ritwikaravind88@gmail.com

Abstract: **Hyper-redundant robots (HRR) have a very large degree of kinematic redundancy, and are analogous in morphology and operation to snakes, elephant trunks, and tentacles. These are highly articulated devices can co-ordinate their internal degrees of freedom to perform a variety of locomotion capabilities that go beyond the capabilities of conventional wheeled and the recently developed legged robots. Snake-like locomotion and worm locomotion is considered here to achieve hyper redundancy. The title is derived from a thorough literature survey on current research trends in robotics. The Robot manipulator will be created from a sequence of link and joint combinations. Redundancy in manipulator design has been recognized as a means to improve manipulator performance in complex and unstructured environments. The mechanical design and implementation of hyper-redundant robots has been perceived as unnecessarily complex. The objective of my project is to implement much simpler mechanical design for attaining a snake-like locomotion without losing much locomotion efficiency and cost effectiveness.**

Key Words: Robot Manipulator, snake-like locomotion, HRR

I. INTRODUCTION

Robot manipulators are created from a sequence of link and joint combinations. The links are the rigid members connecting the joints, or axes. The axes are the movable components of the robotic manipulator that cause relative motion between adjoining links.

The wheel is an amazing invention, but it does not roll everywhere. Wheeled mechanisms constitute the backbone of most ground-based means of transportation. On relatively smooth surfaces, such mechanisms can achieve high speeds and have good steering ability. Unfortunately, rougher terrain makes it harder, if not impossible, for such mechanisms to move. Robot manipulators which has more support than the minimum number of degrees-of-freedom are termed ``kinematically redundant," or simply ``redundant". Redundancy in manipulator design has been recognized as a means to improve manipulator performance in complex and unstructured environments. ``hyper-redundant" robots have a very large degree of kinematic redundancy, and are analogous in morphology and operation to snakes, elephant trunks, and tentacles. (e.g. [3], [8], [18])

A snake robot is a robotic mechanism designed to move like a biological snake. Inspired by the robustness and stability of biological snake locomotion, snake robots carry the potential of meeting the growing need for robotic mobility in unknown and challenging environments (e.g. [8], [18]). These mechanisms typically consist of many serially connected joint modules capable of bending in one or more planes. The many degrees of freedom of snake robots make them difficult to control but provide potential locomotion skills in cluttered and irregular environments which surpass the mobility of more conventional wheeled, tracked, and legged robots.

Snake robots can use their many internal degrees of freedom to thread through tightly packed volumes accessing locations that people and machinery otherwise cannot use. Moreover, these highly articulated devices can coordinate their internal degrees of freedom to perform a variety of locomotion capabilities that go beyond the capabilities of conventional wheeled and the recently developed legged robots. The true power of these devices is that they are versatile, achieving behaviors not limited to crawling, climbing, and swimming. (e.g. [4], [9])

II. PROBLEM DEFINITION

A. *Background and Motivation*

Watching the snake robot moving through a room, it is interesting to observe the surprised reactions of people when it quickly turns towards them. These artificially intelligent machines are the products of human imagination and technical understanding. As the technology advances, the line between living and non-living matter is slowly becoming blurred. Moreover, the snake locomotion helps us to achieve more autonomous nature in controlling the robot rather than other hyper redundant robot (HRR) systems discussed earlier. An HRR snake-like robot can be created as a modular robot using series of links and joints as in [1].

In nature, the snake is one of the creatures that exhibit excellent mobility in various types of terrain. It is able to move through narrow passages and climb on rough ground as in [2], [4], and [10]. This property of mobility is attempted to be recreated in robots that look and move like snakes. Snake-like robots usually have a high number of degrees of freedom (DOF) and they are able to move without using active wheels or legs.

Throughout the world, earthquakes, Tsunamis, hurricanes, landslides, floods etc. are tremendously frightening to all the people whom living at the places like China, Japan, India, USA, Philippines, Indonesia, and others. As we get a closer look to the places like, Haiti, situated in the largest population country in the world, China, in January 12, 2010, it happened for the worst quake in region in 200 years with the death toll at estimated 230,000 and Uttarakhand, Odisha etc. in India during flash floods in August 2013 as in [16], many hundreds were still missing and huge damage to the property. In these situations, deep searching for survivors that trapped in the collapsed buildings must be carried on. Thus, snake-like robots provide a means to replace human by exploring to all hidden and unreachable areas of the incident happened. The advantages of automating snake-like robot will be discussed.

This project presents the prototype snake-like robot and outlines expectations for future development. Snake-like robots may one day play a crucial role in search and rescue operations, firefighting, and inspection and maintenance. The highly articulated body allows the snake-like robot to traverse difficult terrains such as collapsed buildings or the chaotic environment caused by a car collision in a tunnel. The snake-like robot could crawl through destroyed buildings looking for people, while simultaneously bringing communication equipment together with small amounts of food and water to anyone trapped in the shattered building. A rescue operation involving a snake-like robot has been envisioned. Moreover, the snake-like robot can be used for surveillance and maintenance of complex and possibly hazardous areas of industrial plants even inspecting the sewerage system looking for leaks or aiding firefighters. Also, snake-like robots with one end fixed to a base and autonomous systems can be used as a robot manipulator which can reach hard-to-get-to places as in surgical applications [20], (cardiac surgery, endoscopy etc.), nuclear decommissioning, bomb retrieval etc.

B. *Hyper-Redundant Robots (HRR)*

The word "redundant" is used in the context of robotic manipulators to indicate that the number of actuated degrees of freedom exceeds the minimal number required to perform a particular task. For instance, manipulators required to position and orient an object in space needs six actuated degrees of freedom, and so a manipulator with seven or more is redundant with respect to this task. "Hyper-redundant" manipulators are redundant manipulators with a very large degree of redundancy. These manipulators can be analogous in morphology and operation to "snakes," "elephant trunks," or "tentacles." Because of their highly articulated structures, these robots are well suited for operation in highly constrained environments, and can be designed to have greater robustness with respect to mechanical failure than manipulators with a low degree of redundancy. Furthermore, the concept of hyper-redundancy can be generalized beyond manipulators to describe novel forms of robotic locomotion analogous to the motion of worms, slugs, and snakes. Particular hyper-redundant designs have previously been referred to as: "highly articulated," "tentacle," "snake-like," "tensor-arm," "elephant trunk," "swan's neck," and "spine". To my knowledge, the earliest hyper-redundant robot design and implementations date to the late 1960's. Hirose and coworkers have implemented a large number of working high-dof systems. Numerous other authors have suggested hyper-redundant designs or developed hyper-redundant robot mechanisms as in [8]. Many of these designs were driven to some extent by a particular application or operating environmental scenario. Fig. 3.1 exemplifies the three major types of hyper-redundant manipulators: continuous, serial, and cascaded platforms.

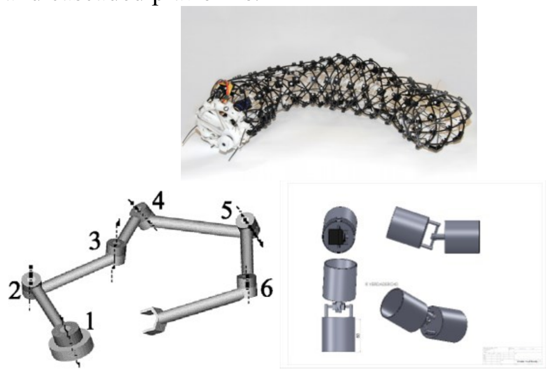

Figure. 1. Types of HRR

The selection of a particular morphology will obviously depend heavily upon the functional and operational requirements of a particular application. Some possible morphology is represented in Fig. 1. To obtain the maximum possible benefit from their highly articulated structures, hyper-redundant manipulators must not only be able to place their end-effectors at a specified location, but must also be able to do things that are difficult or impossible for mildly redundant and kinematically sufficient manipulators, such as:

- Maneuver through highly constrained workspaces grasp objects by completely enveloping them generate peristaltic waves to manipulate objects. These functional requirements arise in the following anticipated application scenarios:
- Operations in highly constrained environments, such as: nuclear reactor steam generator maintenance; chemical sampling in buried toxic waste containers; and medical endoscopy.
- Tentacle-like grasping and object reorientation such as: capturing and de-spinning free floating satellites; replacement of mechanical components in hard to reach areas of an automobile; and complex 'whole arm' manipulation experiments.
- Exploration in complex environments such as: emergency response vehicles in burning or collapsed buildings; and sensor placement in complicated geological formations, such as lava tubes.

In continuous morphologies, actuation is distributed through the robot structure, resulting in the ability to continuously deform the robot's local geometry. Continuous morphologies include robots based on flexible pneumatic and hydraulic actuators and actuator bundles. (e.g. [2])

At first glance continuous manipulators would appear to be more maneuverable and highly articulated. However, their actual implementation can be problematical. Truly distributed actuation (akin to muscles in a tentacle or tongue) is difficult to engineer in practice. Further, pneumatic and hydraulic schemes are not very suitable for locomoting hyper-redundant robots, as it is difficult to carry the pumps, regulators, and other required components in a self-propelled package.

Discrete morphologies are comprised of a finite number of non-distributed actuators. Examples of such systems include serial chain rigid link robots (with revolute and/or prismatic joints). Serial chain rigid link systems are an extension of traditional manipulator design methodology, have a simple kinematic structure, and have simple 'fitting' algorithms. However, this morphology has inherently poor mechanical advantage, making it difficult to build one which can support its own weight when fully extended. In addition, serial chain designs are not very robust with respect to actuator failure.

Variable geometry truss systems have many advantages for practical hyper-redundant manipulator implementation. VGTs were originally studied in the context of large space-structures, such as precision segmented mirrors. More recently, variable geometry truss manipulators (VGTMs) have been proposed and investigated. Traditional VGTs and VGTMs can be differentiated by their actuation requirements. (e.g. [1])

For large space structures, the primary role of the actuators and control system is to control structural resonances. Thus, these systems have actuators with high bandwidth and very little stroke. Conversely, VGTs for robotics applications require actuators with substantial stroke to effect significant changes in the structure geometry. I have chosen VGT geometry for the system because of its inherent mechanical advantage, which is required for several of the experiments in hyper-redundant grasping. (e.g. [3])

C. *Applications*

HRR can be made useful for tasks in tough terrains, to reach every nuke and corner which is not easily accessible and majorly into potentially vulnerable environments. As per recent studies, HRR with snake-like locomotion is suggested to be employed for space missions as they are capable of encountering rough and unpredictable terrains compared to wheeled mechanisms.

Moreover, the versatile properties obtained by mimicking the successfully evolved organisms in nature have made them highly efficient in means of locomotion and accessibility. Snake-like HRR thus pose many advantages in their locomotion in different terrains. Thus it will be easy to use them as amphibious robots, in-pipe robots etc. The various applications of snake-like HRR can be compiled as follows as in [16]

1) *Nuclear Projects:* It helps in the decommissioning of nuclear power plants and other nuclear radiation emitting devices. Repair and maintenance at highly vulnerable radiation sites
2) *Aerospace:* Manufacture and assembly: inside wing boxes, jet engines and ducts
3) *Surface Preparation:* wielding pneumatic sanders for all stages of surface finishing prior to final paint application
4) *Maintenance, Repair and Overhaul*
5) *Automotive Manufacture:* Snake-arm robots allow structures to be assembled in a different way
6) *Security and defense:* Search and rescue, Bomb disposal and counter terrorism
7) *Robotic surgery:* Endoscopy, Neurosurgery

III. HRR DESIGN

The robot snake will be built and programmed which consists of six segments and a head, with each segment being powered by an R/C servo. The segments may alternate in orientation so that the first segment moves in a horizontal motion and the next segment moves in a vertical motion, to achieve Serpentine Locomotion. This sequence repeats itself for all six segments and the head as in [14]. This gives the snake enough flexibility to move its body in a number of different ways in order to achieve locomotion, in much the same way as a biological snake. The robot will be controlled using ATMega2560 Arduino board or Microchip PIC microcontroller. The microcontroller is used to sequence the movement of each of the snake's body sections via servos. The microcontroller also monitors an infrared sensor so that the snake will avoid obstacles as it explores.

The basic aim of the project is to physically implement a Snake-Like Hyper Redundant Robot (HRR). The robot will be implemented without any wheels (active and passive), still capable of moving like a snake. In robotics literature, HRR with snake-like locomotion with passive wheels are actually termed as Snake-like robots whereas those without any wheels are termed as Snake Robots. The major tasks under this work are mechanical design of the robot structure, selection and installation of proper electronic hardware, programming and testing. The robot must have multiple body segments and joints which allow each body segment to move unrestricted relative to the other body segments. The robot will be able to flex, reach, and slither through narrow workspaces with its infinite number of joints configurations. Electronic wise, the robot also must have a good power management which allows the robot to wander freely for long period of time without any cables and travel to farther locations. Another important criteria that mobile robot must have is good control actuations. Good control actuations enable the robot to move efficiently and safely through difficult various obstacles. Programming wise, the robot must have the locomotion of the real biological snake. The robot also must possess the ability to overcome obstacles of certain height limit, or find another alternative routes if it cannot climb over the obstacles. Apart from all these the robot should be simple and cost efficient as in [7].

A. *Robot Body Design*

The robot body will act as a mechanism for achieving the required serpentine locomotion. It should possess good reliability in providing the accessories on board enough safety when facing uncertain terrains. The material selection has to be based on less weight and cost along with characteristics like stable, non-toxic, manufacturability etc. Proper joints are to be selected as the joint movements are rapid and uncertain.

1) *Material Selection:* The material of the segments is fundamental to determine how much force is required to lift and move like a snake robot. As the material used is a heavy, load will lead to large torque requirements, while the light material leads to deformation or break when the force exerted.

In the preparation of the snake like robot, it requires a material that can maintain the strength of stress and tension, like a robot built like a snake must be reliable in performing tasks. In addition, the material should be cheap and easy to get to facilitate the plan for low cost. The material should be good enough to facilitate the manufacturing process and, finally, must be of the required dimensions. Also the shape and sizing of the material should help to achieve proper installation of electronic hardware and wiring.

The candidate materials were Poly Vinyl-Chloride (PVC) tubes and Aluminium sheet metal. Aluminium sheet is selected as the material even though PVC tubes are less costly. The trade-offs given were in ease of shaping, installation of hardware and wiring and comparable weights. A detail comparison is given in the Table 1 below as in [17]

Sl No	Criterion	Aluminium Sheet	PVC Tube
1	Density, g/cm3	2.70	1.3–1.45
2	Tensile Strength, N/mm^2	Approx. 200	Approx. 45
3	Young's Modulus, GPa	70	3.37
4	Melting Point, ^{O}C	160	660

Table.1 COMPARISON OF MATERIAL PROPERTIES

2) *Type of Segment (Links and Joints):* As the selected material is Aluminium sheet metal the links and joints are modeled for ease of fabrication. To improve the frictional coefficient, the joints are properly greased or lubricated. The links are made for easy installation of servos and wiring required. All segments are fabricated using conventional sheet metal fabrication tools and techniques. Detailed drawings are illustrated below. The head and tail segments are designed to achieve a snake-like appearance.

The links and joints are designed for ease to attach the drive system like servomotors and wiring. The head and tail sections are attached to carry sensor and processor modules. The tail section also needs to provide support to the HRR while in movement.

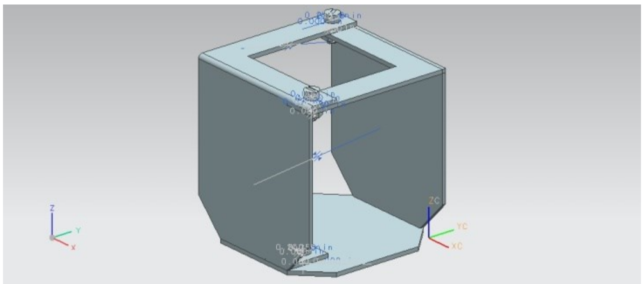

Figure. 2. Assembly of Part A & B (Segment)

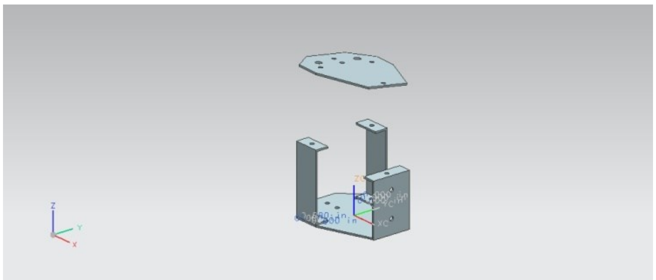

Figure. 3 Assembly of Head Section

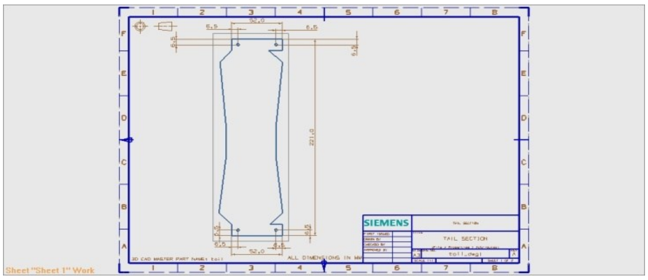

Figure. 4 Detailed Drawing for Tail Section

3) Assembly: The assembly of the segments is done using locking nuts. Provision for servo horn attachments are provided in each segment. The segments are arranged so as to obtain alternate horizontal and vertical movements as shown in Fig. 5 below from [14].

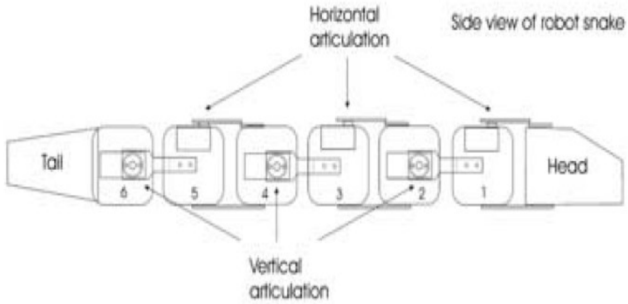

Figure 5 Assembly of Snake-like HRR

4) Servomotor Torque Calculation: The torque requirement for the prototype was calculated after the fabrication of model. The weight of segments were found out and the calculation was made for Torque, T = F.r, where F is the weight to be lifted and r be the distance. From the calculations, maximum torque required for servomotors is 8.16 kg-cm. The servo can be selected to meet the average torque of 4 kg-cm.

B. Electronic Architecture

Engineering the hardware architecture refers to the identification of physical components of a system and their interrelationships. This disclosure, often called a model for hardware configuration enables hardware developers to understand how the components fall within system architecture and provides designers of software components important information necessary for software development and integration. A clear definition of a hardware architecture will enable these different traditional engineering disciplines (eg, electrical and mechanical engineering) to work more effectively together to develop and to produce new machines, components and devices.

1) Micro Controller Selection: All microcomputer systems, irrespective of their complexity, are based on similar building blocks. These are shown in figure below and consist of the following:

- CPU - the part that does all logic and arithmetic functions
- RAM - storage for programs and/or program variables
- ROM - read-only parts of programs
- I/O - connection to external devices

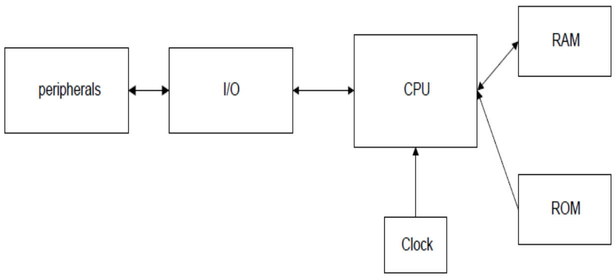

Figure. 6. Micro Controller Architecture

Selecting the right microcontroller for a product can be a daunting task as in [15]. Not only are there a number of technical features to consider, there are also cases of business issues such as cost and delivery that can cripple a project. Before any thought is given to the microcontroller, the work of the high level system block diagram and flowcharting and only then is there enough information to start making a rational decision about selecting microcontroller. In present day robotics, PIC Microcontrollers and Arduino Microcontrollers are widely used. PIC Microcontrollers are from MICROCHIP Technology whereas Arduino Microcontrollers are open source microcontroller platforms and related ecosystems which enable people (developers, artists etc) to integrate various hardware and software components in a fascinating and creative way. The potential candidates for microcontrollers here are ATMega2560 Arduino board and Microchip PIC microcontroller.

1) Servomotor Selection: To mimic the locomotion of a biological snake, there should be a mechanism involved. One possible mechanism would be the DC servo motor. Servomotors are widely used in robotic applications than any type of electric motors as they are capable of achieving closed loop control. A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. (e.g. [5], [11])

Servomotors are not a different class of motor, on the basis of fundamental operating principle, but uses servomechanism to achieve closed loop control with a generic open loop motor. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

Motor and drives selection process as well as the mechanical transducer system design are a very important step. Servo is selected based on the torque calculations performed. For HRR, all segments will be provided with servomotors meeting maximum torque required. Otherwise, non- optimal choices leads to poor system performance and increased installation and maintenance costs. The first step is to choose an appropriate setting to conduct a review of the load. The purpose of the expenditure required to meet the work load of the motor , square (RMS) of torque capabilities mean enough speed , peak torque , and to root . Equally important is to choose the type and amount of electricity to meet the needs of the drive.

When choosing a driver, a peak and RMS current and voltage to the motor winding can provide to meet the needs of both. When the driver selects a task, the interface should be checked. Some examples of the analog speed command interface, and type AB is the pulse shape of the pulse shape and pulse & direction. Some drives position control function is integrated into the amplifier. Considering the type of device and the concept of a working system, input / output (I / O), and other features to be made. The DC power supply provides power amplifier based on a task. High power output rating of a power supply to run at the same time all work must be equal to the average driver. The figure below illustrates servo control architecture. [5]

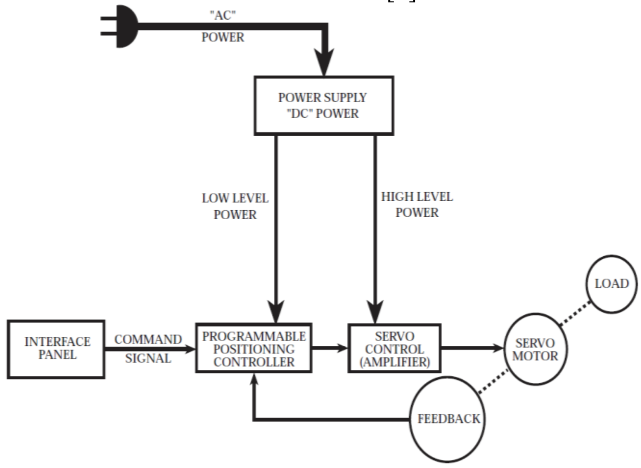

Figure.7. Servo Control Architecture

2) Sensors: For the efficient control and guidance of the snake like locomotion proper sensors are to be selected. The major requirement is for path guidance and obstacle avoidance. For this an Infrared (IR) sensor will help to get the required input. A video camera with night vision is also under consideration as it helps to acquire real time environment of robot movement. Proper sensor modules have to be attached at the head and tail ends for hassle free navigation.

Sensors can also be included to give feedbacks on the orientation angles and contact forces on the robot segments along with application oriented sensors like temperature sensors, oxygen level sensors etc.

The electronic architecture thus involves all the system components along with other accessories and wiring required for the motion control of the robot.

C. Software and Programming

The software for robot programming will be based on the microcontroller to be used (e.g. [12], [13], [19]). PIC programming or Arduino programming will be used in their respective interfaces or using C++ programming. The flow chart for program algorithm is discussed as below.

As per the mechanical design and electronic architecture, 3 servos are to be programmed for horizontal movement and other alternate 3 has to be programmed for vertical displacement. A basic flow chart representation is as shown in Fig. 8. In this flow chart POSITION_RIGHT, POSITION_LEFT, SERVO_RIGHT, SERVO_LEFT etc. are sub programs in the software to specify the movement for servos in sequence. The program is to be updated with reverse movement of the robot when get into a closed enclosure. This enables proper retrieval of the robot from pot-holes. The initial decision variable as while “S” will act as switch ON/OFF function. Servo sequences to be executed are to be fed properly by considering the response timings and rotation angles, into the micro controller for precise movement.

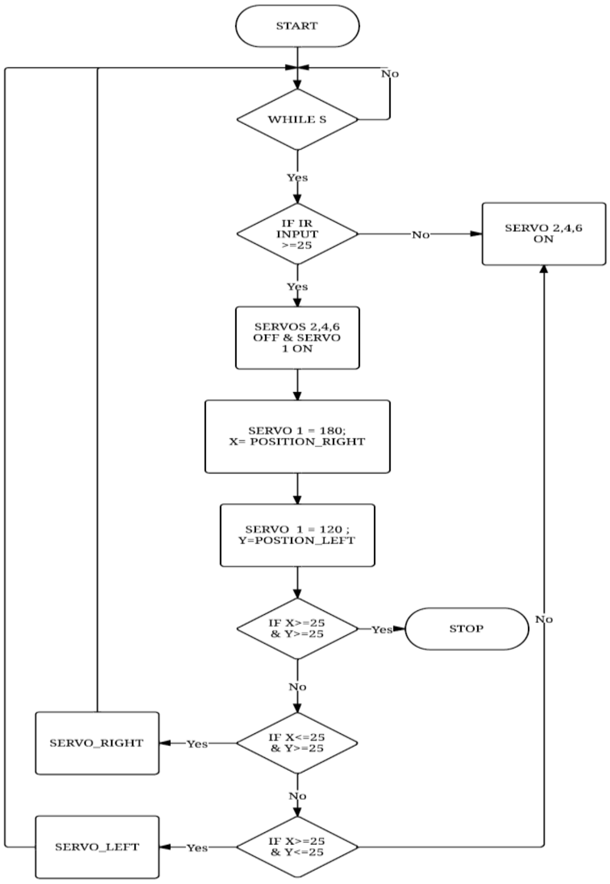

Figure. 8. Flow-Chart for HRR Program

Arduino Program for Robot Locomotion:

```
#include <Servo.h>
#include<stdlib.h>
  char Recdata;
  char i;
  int Distance_value=0;
  int obstacle_threshold=200;
  char servo1_angle=90;
  char servo2_angle=110;
  char servo3_angle=90;
  char servo4_angle=100;
  char servo5_angle=40;
  char servo6_angle=70;
  Servo Segment1_servo;
  Servo Segment2_servo;
  Servo Segment3_servo;
  Servo Segment4_servo;
  Servo Segment5_servo;
  Servo Segment6_servo;
  void Forward(void);
  void Stop(void);
  void Right(void);
  void Left(void);
  void obstacles_avoidance(void);
  void obstacle();
  unsigned int adc_threshold;
void setup()
{
   Serial.begin(9600);
   Segment1_servo.attach(2);        //Head of snake robot connected to the digital pinout 2
   delay(100);
   Segment2_servo.attach(3);      //Segment2 of snake robot connected to the digital pinout 3
   delay(100);
   Segment3_servo.attach(4);      //Segment3 of snake robot connected to the digital pinout 4
   delay(100);
   Segment4_servo.attach(5);      //Segment4 of snake robot connected to the digital pinout 5
   delay(100);
   Segment5_servo.attach(6);      //Segment5 of snake robot connected to the digital pinout 6
   delay(100);
   Segment6_servo.attach(7);   //Tail segment of snake robot connected to the digital pinout 7
   delay(100);
  Segment1_servo.write(servo1_angle);
  delay(500);
  Segment2_servo.write(servo2_angle);
  delay(500);
  Segment3_servo.write(servo3_angle);
  delay(500);
  Segment4_servo.write(servo4_angle);
  delay(500);
  Segment5_servo.write(servo5_angle);
  delay(500);
  Segment6_servo.write(servo6_angle);
  delay(500);
}
void loop()
{
if (Serial.available() > 0)
  {
  Recdata=Serial.read();
  }
   if(Recdata=='F')
     {
    adc_threshold = analogRead(0);
if( ( adc_threshold > obstacle_threshold ) )
             {
     obstacles_avoidance();
         }
     else
     {
     Forward();
     }
     }
   if(Recdata=='S')
     {
     Recdata='\0';
     Stop();
     }
}
void Forward(void)
  {
    for (i=0;i<=45;i++)
     {
     Segment6_servo.write(servo6_angle--);
     delay(3);
```

```
   }
  for (i=0;i<=45;i++)
   {
   Segment4_servo.write(servo4_angle++);
   delay(3);
   }
  for (i=0;i<=55;i++)
   {
   Segment6_servo.write(servo6_angle++);
   delay(3);
   }
  for (i=0;i<=45;i++)
   {
   Segment2_servo.write(servo2_angle++);
   delay(3);
   }
  for (i=0;i<=55;i++)
   {
   Segment4_servo.write(servo4_angle--);
   delay(3);
   }
  for (i=0;i<=55;i++)
   {
   Segment2_servo.write(servo2_angle--);
   delay(3);
   }
   servo6_angle=70;
   Segment6_servo.write(servo6_angle);
   servo4_angle=100;
   Segment4_servo.write(servo4_angle);
   servo2_angle=110;
   Segment2_servo.write(servo2_angle);
 }
void Left(void)
{
for (i=0;i<=35;i++)
    {
     Segment1_servo.write(servo1_angle++);
     delay(10);
    }
  for (i=0;i<=35;i++)
    {
     Segment3_servo.write(servo3_angle++);
     delay(10);
    }
   for (i=0;i<=35;i++)
    {
     Segment1_servo.write(servo1_angle--);
     delay(15);
    }

   for (i=0;i<=45;i++)
    {
     Segment5_servo.write(servo5_angle--);
     delay(15);
    }
   for (i=0;i<=35;i++)
    {
     Segment3_servo.write(servo3_angle--);
     delay(15);
    }
      for (i=0;i<=45;i++)
    {
     Segment5_servo.write(servo5_angle++);
     delay(15);
    }
}
void Right()
{

 for (i=0;i<=45;i++)
    {
     Segment1_servo.write(servo1_angle--);
     delay(10);
    }
   for (i=0;i<=35;i++)
    {
     Segment3_servo.write(servo3_angle--);
     delay(10);
    }
    for (i=0;i<=45;i++)
    {
     Segment1_servo.write(servo1_angle++);
     delay(15);
    }
   for (i=0;i<=45;i++)
    {
     Segment5_servo.write(servo5_angle++);
     delay(15);
    }
    for (i=0;i<=35;i++)
    {
     Segment3_servo.write(servo3_angle++);
     delay(15);
    }
       for (i=0;i<=45;i++)
    {
     Segment5_servo.write(servo5_angle--);
     delay(15);
    }
}
void Stop(void)
{
    servo1_angle=90;
    servo2_angle=110;
    servo3_angle=90;
    servo4_angle=100;
    servo5_angle=40;
    servo6_angle=70;

   Segment1_servo.write(servo1_angle);
   delay(100);
   Segment2_servo.write(servo2_angle);
   delay(100);
   Segment3_servo.write(servo3_angle);
   delay(100);
   Segment4_servo.write(servo4_angle);
   delay(100);
   Segment5_servo.write(servo5_angle);
   delay(100);
   Segment6_servo.write(servo6_angle);
   delay(100);
}
void obstacles_avoidance(void)
{

unsigned int peak_left_threshold;
unsigned int peak_right_threshold;
```

```
adc_threshold = analogRead(0);

if( ( adc_threshold > obstacle_threshold ) )
        {
        delay(7);
        peak_left_threshold  = adc_threshold;
        peak_right_threshold = adc_threshold;

                for(i=0 ;i < 30;i++ )
                        {
Segment1_servo.write(servo1_angle++);
                        delay(7);
                        adc_threshold =
analogRead(0);
                        if( adc_threshold >
peak_left_threshold +10)
                                {

        peak_left_threshold = adc_threshold;
                                }
                        }
                for(i=0 ;i < 30;i++ )
                        {

        Segment1_servo.write(servo1_angle++);
                        delay(7);
                        }

                for(i=0;i < 30;i++ )
                                {

        Segment1_servo.write(servo1_angle--);
                        delay(7);
                                adc_threshold =
analogRead(0);

        if( adc_threshold > peak_right_threshold+10 )
                                        {

        peak_right_threshold = adc_threshold;
                                        }
                                }
                for(i=0 ;i< 30;i++ )
                                {

        Segment1_servo.write(servo1_angle--);
                                delay(7);
                                }

                        if( peak_left_threshold >
peak_right_threshold )
                                        {
                Right();Right();

                                        }
                        if( peak_right_threshold >
peak_left_threshold )
                                        {

                Left();Left();
                                        }
        peak_left_threshold = 0;
        peak_right_threshold = 0;
   }
}
```

IV. Conclusion and Future Work

A Hyper redundant robot manipulator with snake like locomotion is proposed and the design is carried out. The bottleneck in this project is in programming and integration of electronic architecture which requires much optimization process. This is to precisely navigate the HRR in different terrains. Servo motors selected are based on the maximum torque requirement, which enable maximum performance and redundancy even if any of the motors fail during operation. Robot programming is based on servomotor activation sequences required to achieve snake like locomotion.

Future work can be the up-gradation of robot program which enables the robot to get out of pot-holes [21] or fully surrounding obstacles using a reverse program. Sensors can be improved to night vision camera along with orientation and contact force sensors. The designed model can be modified for specific requirements just by modifying the servomotor capacity and robot body structure [22]. Shape memory alloy like Nickel- Titanium wires can be used to considerably smaller sizes.

REFERENCES

[1] Abdelghani Chibani, Yacine Amirat, Samer Mohammed, Eric Matson, Norihiro Hagita and Marcos Barreto (to be published)"Ubiquitous robotics: Recent challenges and future trends", Robotics and Autonomous Systems, 2013. Pages 1162-1172

[2] Alexander S Boxerbaum, Kendrick M Shaw, Hillel J Chiel and Roger D Quinn, "Continuous wave peristaltic motion in a robot", IJRR, 2012, Pages 302-318

[3] Annisa Bt Jamali, Md. Raisuddin Khan and Mozasser Rahman, "Design and Control of Variable Length Hyper Redundant Robot", Procedia Engineering, 2012, Pages 1105-1113

[4] Avishai Sintov, Tomer Avramovich and Amir Shapiro, "Design and motion planning of an autonomous climbing robot with claws", Robotics and Autonomous Systems, 2011, Pages 1008-1019

[5] Dal Y. Ohm, "SELECTION OF SERVO MOTORS AND DRIVES" Drivetech, Inc., Blacksburg, Virginia, 2006, Drivetech Inc., Technical Articles

[6] F.L. Chernousko, "Modelling of snake-like locomotion", Applied Mathematics and Computation, 2005, Pages 415-434

[7] James Trevelyan, "Simplifying robotics - A challenge for research" Robotics and Autonomous Systems, 1997, Pages 207–220

[8] Liljebäck, K.Y. Pettersen, Stavdahl, and J.T. Gravdahl, "A review on modelling, implementation, and control of snake robots", Robotics and Autonomous Systems, 2011,Pages 29-40

[9] Mohd Safuwan Mohd Jasni, Reza Ezuan Samin and Babul Salam Kader Ibrahim, "Biological Inspired Inspection Underwater Robot (SNAKEY)", Procedia Engineering, 2012, Pages 1058-1064

[10] Nur Shahida Roslin, Adzly Anuar, Muhammad Fairuz Abdul Jalal and Khairul Salleh Mohamed Sahari, "A Review: Hybrid Locomotion of In-pipe Inspection Robot", Engg. Procedia, 2012, Pages 1456-1462

[11] Tianlin Song, Ling Zao and Xuanxuan Shen, "The Research on Remote Control Robot Snake Driving by Servo", Procedia Engineering, 2011, Pages 762-766

[12] Xiaodong Wu and Shugen Ma, "CPG-based control of serpentine locomotion of a snake-like robot", Mechatronics, 2010, Pages 326-334

[13] Zijian Tian, Liya Zhang and Wei Chen, "Improved algorithm for navigation of rescue robots in underground mines" Computers and Electrical Engineering, 2013, Pages 1088-1094

[14] Karl Williams, "Amphibionics", Mc Graw Hill, 2003

[15]http://www.edn.com/electrical-engineer-community/industry-blog/4411480/1/10-steps-to-selecting-a-microcontroller

[16] http://en.wikipedia.org/

[17] http://www.engineeringtoolbox.com

[18] www.snakerobots.com

[19] David Johan Christensen, Ulrik Pagh Schultz and Kasper Stoy, "A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots" Robotics and Autonomous Systems, 2013, Pages 1021-1035

[20] Elif Ayvali, Chia-Pin Liang, Mingyen Ho, Yu Chen and Jaydev P Desai, "Towards a discretely actuated steerable cannula for diagnostic and therapeutic procedures", IJRR, 2012, Pages 588-603

[21]Filippo Sanfilippo ,Jon Azpiazu ,Giancarlo Marafioti et.al., " Perception-Driven Obstacle-Aided Locomotion for Snake Robots: The State of the Art, Challenges and Possibilities", Applied Sciences 2017, 7(4), page 336.

[22] Zhenshan Bing, Long Cheng, Guang Chen, Florian Röhrbein, Kai Huang and Alois Knoll, " Towards autonomous locomotion: CPG-based control of smooth 3D slithering gait transition of a snake-like robot" Bioinspiration & Biomimetics April 2017, Volume 12, Number 3