RESEARCH ARTICLE OPEN ACCESS

ROBRA – The Snake Robot

James M.V, Nandukrishna S, Noyal Jose, Ebin Biju

Department of Computer Science and Engineering
Toc H Institute of Science & Technology, Ernakulam
Kerala – India

ABSTRACT

Robra - A snake-like robot which, provides the locomotion of a real snake. The shape and size of the robot depends on application. Robra can avoid obstacles by receiving signals from sensors and move with flexibility on terrain surfaces. Multiple joints increases its degree of freedom. It can quickly explore and navigate foreign environments and detect potential dangers for direct human involvement.

Robra uses PIR() to detect presence of humans in an environment like civilians trapped in collapsed buildings during natural disasters. It can move through crevices and uneven surfaces. It detects obstacles by using Ultrasonic range movement sensors. This provides noncontact distance between the obstacles to detect and avoid collision with them. Electro chemical sensors are used for detection of poisonous gases. Robra has an onboard camera which records and send the video data in real time via WiFi. Robra has microphones for the survivors to communicate and inform their status incase of rescue aid. Thus Robra covers wide real time applications like surveillance, rescue aid etc.

Keywords:- Robra, Snake,

I. INTRODUCTION

In the past two decades it is estimated that disasters are responsible for about 3 million deaths worldwide, 800million people adversely affected. Urban Search and Rescue may be needed for a variety of situations, including earthquakes, hurricanes, tornadoes floods, fires, terrorist activities, and hazardous materials (hazmat) accidents. If a person is found extrication can take even longer. Hence, there is the need of a Smart robot which can crawl through crevices and Find people. Snake-like robot is a biomorphic hyper-redundant robot that resembles a snake. The shape and sizes of the robot depends on its own application, different application require different size and shapes, for example, for research and rescue purpose the robot should be as thin as possible to enable it to do the tasks at a narrow place. In order to make the robot function and move like a real snake, Robra is constructed of multiple joints which enable it to have multiple degree of freedom. Multiple degree of freedom gives it the ability to flex, reach and approach a huge volume in its workspace with infinite number of configurations.

II. LITERATURE SURVEY

Smart robots are also equipped with multiple sensors for precise movement and sensing. Some of them are mentioned as follows.

Chimp - Designed by the Tartan rescue team from the Carnegie Mellon University's National Robotics Engineering Center, CHIMP is a four-limbed human-size robot that, when standing, is 5-foot-2-inches tall and weighs about 400 pounds. It is primarily not a walking robot, but one designed to move on tank-like treads. When it needs to operate power tools, turn valves, or otherwise use its arms. The robot's long front arms act have a wingspan almost 10 feet, giving it an ape-like appearance.

Momaro - This robot has been specifically designed by the team NimbRo Rescue from the University of Bonn in Germany, to the requirements of the DARPA Robotics Challenge. It consists of an anthropomorphic, or human-like, upper body fitted on a flexible hybrid mobile. Momaro has four legs which end in pairs of directly-driven, steerable wheels. The upper body consists of two adult-sized anthropomorphic arms with seven degrees of freedom each. Momaro is equipped with a sensor head consisting of a continuously rotating 3D laser scanner, which produces a spherical field-of-view, eight color cameras with an omnidirectional field-of-view, three full HD color cameras for a panoramic operator view, and a top-down, wide-angle camera.

Helios - A central component of the system is an inverse kinematics (IK) engine that computes the necessary robot joint angles to achieve some real-world objectives such as positioning the robot's hand at a given point in space. Balancing, walking, and manipulation are accomplished with a controller.The team has also developed an estimation tool which combines information from the robot's joint position sensors, its onboard accelerometer and gyroscopes, and its laser range-finder to determine the robot's state and enable accurate, repeatable foot placement, and even allow the robot to accurately track its state without any contact with the ground. The custom user interface integrates information from the robot's state estimator, the laser scanner, and the robot's onboard cameras and force sensors to give a human operator a detailed view of the world.

III. METHODOLOGY

The Robra is a serpentine robot which can traverse different terrains then perform variety of important functions such as avoid obstacles, locate injured people and use sensors to perform different crucial functions.

1. Obstacle avoidance: Robra has ultrasonic sensors to detect and measure distance between itself and the obstacle. This distance and information about obstacle in its vicinity is used to find the optimal path to avoid the obstacle. In some cases the path that is found to be optimal would not have dead ends, when Robra faces such a situation it backtracks and find a new path. This is done repeatedly until it finds the best path. Ultrasonic sensors with the help of yawing motion of head segment senses the obstacle in front of it

2. Locating injured people: Onboard PIR(passive infrared) sensors detects heat sources. By adjusting the sensitivity a clearer picture can be developed by eliminating all the extreme temperature differences, this further helps in identifying and pin pointing temperature in the range of human body temperature. By programming to filter out the range of temperature that falls under body temperature of human body Robra is able to detect and locate injured peoples.

3. Other sensors: Robra has sensors with the following functions:
Metal detector: Metal detectors help in finding metal underneath the surface. This application is added to Robra for detecting mines.
Gas sensors: Gas sensors detect the presence of methane, carbon monoxide in the environment. This can help in navigating congested areas like tunnels.

IV. IMPLEMENTATION

HARDWARE REQUIREMENTS

- 1 Arduino UNO (with optional sensor shield)
- 24 channel servo driver
- servos
- Ultrasonic sensors
- PIR Sensor
- 1 Lithium Ion Battery
- Several sensor cables and connectors
- 1 5AA battery holder with barrel plug
- Raspberry PI Camera
- Various nuts, bolts, wire clips

SOFTWARE REQUIREMENTS

- latest version of the Arduino IDE
- XCTU
- Raspian OS
- VNC Viewer

Input and Output pins used in snake like robot.

XAMPP : Server Setting- XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Inputs/Outputs Pin	Description
RA5 (AN4)	Used as the analog-to-digital converter channel (sensor).
RA7 (OSC1)	For the external clock source (20 Mhz crystal is used).
RA6 (OSC2)	For the external clock source (20 Mhz crystal is used).
RC2 (CCP1)	PWM output for DC motor.
RD1	The output indicator LED for sensor.
RB1-7	The output signals (duty cycle) for servo 1-7 respectively.
RD7	The output signals (duty cycle) for servo 8.
RB0	For switch (To start the robot without using wireless device).
RC7 (RX)	Receive information from user through wireless device.
RC6 (TX)	Transmit information to user through wireless device.

The snake equation

There are four types of movement for a snake, which are the serpentine movement, rectilinear movement, concertina movement, and side-winding movement. However, for this project, only the serpentine movement is applied as it is the most frequently used form of locomotion by a snake .For the serpenoid curve analysis, it is introduced by Hirose. In this project, the serpenoid curve is used as the basic body movement of the snake robot. While, the serpenoid curve can be modeled with the following equations (Chang & Chen 2008):

$$x(s) = \int_0^s \cos(\zeta_\sigma) d\sigma$$
$$y(s) = \int_0^s \sin(\zeta_\sigma) d\sigma$$

$$\zeta_\sigma = a\cos(b\sigma) + c\sigma$$

$$\phi_i = \overset{h}{\underset{n}{\alpha}} sin(\omega t + (i-1)\beta) + \gamma, (i = 1, ..., n-1)$$
$$\gamma = -\frac{c}{n}$$

$c = 0$

With the values of each variable show above, the parameters α, β, and γ can be calculated by the microcontroller as well. Sincerobra only has 4 segments, hence, it only requires three equations for the joints (three joints). Below shows the equations for calculating the angle at each joint of the robot in programming code:

proprietary web development tool from Adobe Inc. It was created by Macromedia.

```
ang1=(alpha*sin(t)+gamma+3950);
ang2=(alpha*sin(t+1*beta)+gamma+3950);
ang3=(alpha*sin(t+2*beta)+gamma+395
0);
```

n = 4

The number 3950 in the equations represent the decimal value for timer to produce the duty cycle that position the servo motor to 90 degree. In order to make the reference axis centred at 90 degree, the number 3950 is added into the equations. Besides that, the gamma in the equations is used to determine the direction of the robot. If gamma is decreasing, robra will turn to the left direction, or turn to the right direction when the gamma is increasing.

The polling and interrupt method

In robra, both polling and interrupts method are applied. The polling method is used to transmit data to the user, while the interrupts method is used by the timer (Timer 1) to produce the particular duty cycle for servo motors and to receive data from user via serial communication. In polling method, the microcontroller will continuously monitors the status of the TXIF, when the transmit interrupt's flag is low, means the data in TXREG already transmitted to the user via the wireless device, so it will insert the next data to the TXREG again, and the polling will keep repeated until all of the data are transmitted to the user. In interrupt method, whenever the wireless device (to receive data from user) or servo motors needs the microcontroller's service, they will notify it by sending an interrupt signal. Upon receiving the interrupt signal, the microcontroller stops whatever it is doing and serves the device. The interrupt service routine (ISR) are

$a = 45$ degree

$b = 3\pi$

required in interrupts method, because the microcontroller will go to the particular ISR with respect to which interrupt signal it received, and process the command inside the ISR.

The high priority interrupts (received interrupt).

The received interrupt (serial communication) is set to high-priority interrupts. Thus, when the interrupt flag of the received interrupt, RCIF is raised after the entire frame of data, including the stop bit, is received, the microcontroller will be directed to the rom location at 0x0008 and process the ISR of the received interrupt. As high-priority interrupt, the microcontroller will process the data received from the user first and stop executing the current program even for the ISR of timer (low-priority).The received interrupt is used to process the information from the user via the wireless device.

The low priority interrupts (Timer 1 interrupt)

The Timer 1 interrupt is low-priority interrupt, it will direct the microcontroller to the ISR for low priority interrupts once the TMR1IF is raised to indicate that the Timer 1 is overflowed. However, the timer's interrupt will not be process until the finish process of ISR for high-priority interrupts by the microcontroller, due to the lower priority. Timer 1 is used to determine the duty cycle of the signal to servo motors over time. The details of using Timer 1 to control the position of servo motors will be discussed on the sub-chapter for timers.

Snake movement programming (with assist of timers)

The snake locomotion of robra is serpentine movement. Thus, in order to program the microcontroller to move the servo motors in serpentine curve, the equations mentioned above are used in the snake movement programming. In order to provide the serpentine curve by the servo motors, the particular duty cycle PWM signals are required to send to the servo motors in every 20ms. Thus, the method of using the timer interrupt is applied to robra, and Timer 1 is used for this method. Besides the Timer 1, the Timer 0 is used in robra as well, the purpose of using Timer 0 is to help the robot in differentiating obstacles and to make the decision to overcome it.

Timer 1 programming

Since the servo motors get the pulse at every 20ms, so Timer 1 is program to count the time and interrupt the microcontroller in every 20ms. Timer 1 is a 16-bit timer, and it consist of two bytes, which referred to as TMR1L (Timer 1 low byte) and TMR1H (Timer1 high byte). (McKinlay et al., 2007) T1CON register is introduced to select the option for Timer 1. In this project, a binary value of 0b00010000 is loaded into T1CON register to set Timer 1 to have 1:2 prescale, and using internal clock source (crystal frequency). After setting up the Timer 1, the initial count values are required for TMR1L and TMR1H

Sensor programming

Sensor is important for the automation of a robot, in order to give robra the ability to make decision itself, a sensors are installed on it. The sensor used for detecting obstacles is the ultrasonic sensor: MaxSonar- EZ1 sonar range finder, it can sense the object from 0 inch to 254 inches, with two sensing resolution, which is 1 inch resolution at range from 6 inch to 254 inches and 6 inches resolution from 0 inch to 6 inch. Besides that, the interface output format for this sensor can be in pulse width output, analog voltage output, and serial digital output (serial communication). However, the only output format used in this project is analog voltage output, in which the output voltage is range from 0 volt to 2.55 volts with the scaling factor of 10mV per inches. Robra is required to make decision when it senses an object or obstacle within the range of 35cm from the robot, which is the optimize range to let the robot to respond. Since the output format for the sensor is in analog voltage form, and the PIC microcontroller will not respond well to the variation of the analog voltage input, so robra has ADC programming as well.

1 ADC programming

In the ADC programming, some of the major characteristics of the ADC are as follows:

-<u>Resolution</u>

The ADC has *n*-bit resolution, where *n* can be 8, 10, 12, 16, or even 24 bits. The higher-resolution ADC provides a smaller step size, where step size is the smallest change that can be discerned by an ADC. However, the ADC in PIC18F microcontroller is 10-bit ADC, so the number of steps is equal to 1024.

<u>-Conversion time</u>

In addition to resolution, conversion time is another major factor in judging an ADC. Conversion time is defined as the time it takes the ADC to convert the analog input to a digital (binary) number. For the ADC in PIC18F microcontroller, the conversion time is dictated by the internal clock source (crystal frequency).

<u>Reference voltage</u>

Reference voltage is one of the important factors that determine the step size of the ADC output. Since the ADC in PIC18F microcontroller is 10-bit resolution, so the step size of the output can calculate if the Vdd of PIC18 is used as the reference voltage (5V).

Figure 3.2 show the method of switching the channel with respect to the input from user, where ‘1’ is to turn on the ADC, and ‘2’ is to turn off the ADC.

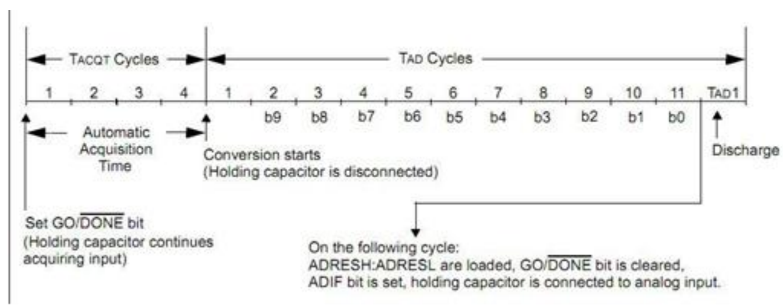


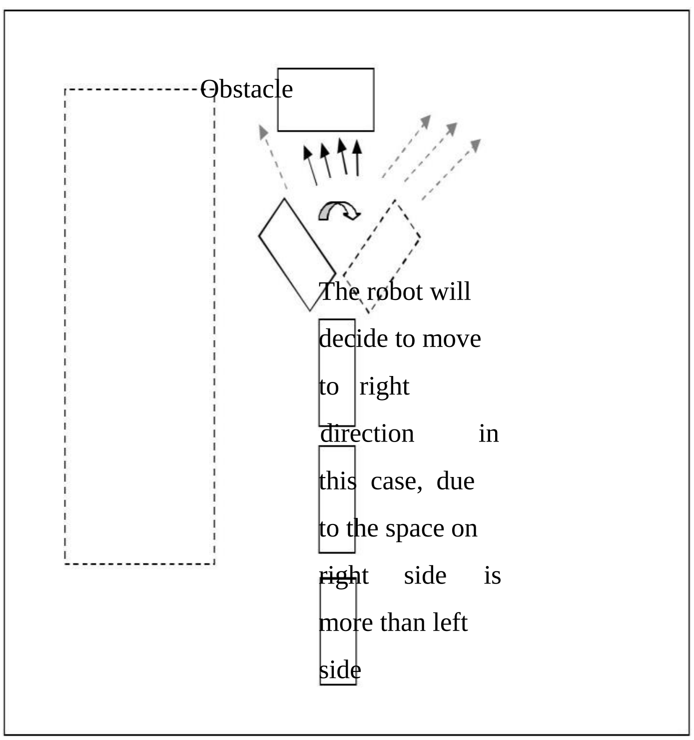
Figure 5.1: The data acquisition and conversion cycle. (Microchip, 2008)

Method of snake-like robot used to avoid obstacle

There are two methods for snake like robot to avoid obstacle proposed in this project, where the first method is to stop and centre the snake like robot, then sense the obstacle form, where the second method is to sense the way without obstacle, then move to the particular direction.

First method for obstacle avoidance

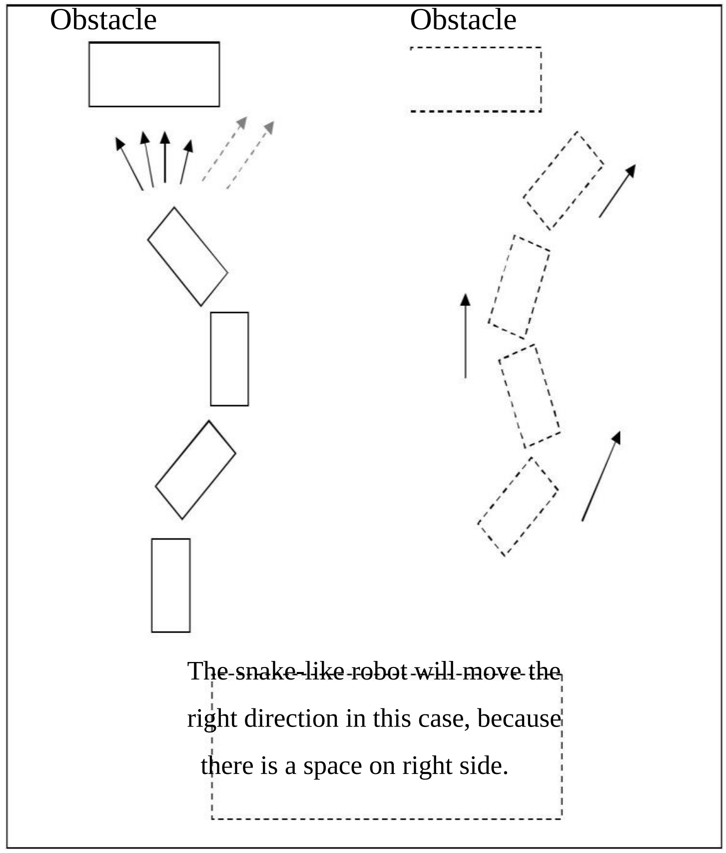
For the 1st method, the snake-like robot will stop and centre all the segments once the sensor detects any obstacle within the desired range. After a certain delay, the head of the snake-like robot (1st segment) will turn to end of left (45 degree to the left). Then, the 1st segment will move step by step with a certain step size of angle, until it reaches the end of right. At the same time, the microcontroller will remember the angle once the sensor detects the obstacle, and then accumulate the steps that the obstacle still within the range (obstacle form) to calculate the size of obstacle, and determine the end of the obstacle, then the snake-like robot will determine the direction to move. If there is no outside path for snake-like robot, then the robot will go to the climbing stage to overcome the obstacle.



First method of obstacle avoidance

2 Second method for obstacle avoidance

The 2nd method for obstacle avoidance is less time consuming, because the snake-like robot will not stop and have delay as 1st method do, the snake-like robot will sense the solution path while it is moving. Once there is a space after an obstacle, then the snake-like robot will turn to the direction



Second method of obstacle avoidance.

Timer 0 programming

Since the Timer 0 is used to help robra to make a decision, so the interrupt for Timer 0 is disable. The polling method is used for Timer 0, the microcontroller will keep monitor the interrupt flag bit of Timer 0, and break the loop if Timer 0 overflowed before the robot can sense the solution way to avoid the obstacle.

V. FUTURE WORK

The robot could easily be developed further with future work. While designing and constructing the robot there has always been a thought of keeping the construction simple so that everything is easy to change and space to improve in the future. Each segment is therefore independent of the others and easily connected mechanically with a few standard bolts and electrically in contact. Incase if one segment breaks down, it could easily be removed and the robot would work without a problem. It is easy to replace the head with a new one if for example, more sensors is to be used for developing more complex maneuvers. Besides improving the steering code and configuration of the already existing abilities, the following features could be implemented without reconstruction of the whole robot. More segments could be installed so that the segments could make up more than one period of a wave or have smoother shape since the angular offset δ can be smaller. This could result in a more snake-like motion as well as a stronger robot because more propelling force could be generated. Due to restrictions set early in the project, no other movement than lateral undulations have been examined. This could be a potential area of expansion for future project. Movements like side winding, where the snake moves sideways could be implemented. The robot is absolutely capable of performing this maneuver. If the robot were to be developed further this could be a great place to start.

VI. CONCLUSION

In developing the current modular snake robot design- “Robra”, we considered several factors. The robot must satisfy various constraints when confronted with the challenges of high-level gaits. The architecture design must consider size, weight and power while producing the necessary torque at every joint. Even while juggling with these constraints, the design maintains a very high level of reliability. This has resulted in a very versatile robot that can function in a wide variety of environments. The development of the Super Servo has been an integral part of the achievements of Robra. While the current implementation has been quite successful, more development is necessary to achieve a fully functional and robust robot. Future designs will most likely no longer make use of hobby servos due to the lack of reliability in the manufacturing process. Using higher voltage motors will eliminate the need for such high current switching supplies. Eliminating dependence on hobby servos allows a more flexible mechanical design as the module shape and cross section will no longer be constrained by the shape of the hobby servo.

REFERENCE

[1] Junyao Gao; Xueshan Gao; Wei Zhu; Jianguo Zhu; Boyu Wei; , "Design and research of a new structure rescue snake robot with all body drive system," Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on , vol., no., pp.119-124, 5-8 Aug. 2008 .

[2] Hua Liu; Guozheng Yan; Guoqing Ding; , "Research on the locomotion mechanism of snake-like robot," Micromechatronics and Human Science, 2001. MHS 2001. Proceedings of 2001 International Symposium on , vol., no., pp.183-188, 2001

[3] http://mcu-programming.blogspot.com/2006/09/servo-motor-control.htm

[4] Maxbotix. (2005). Retrieved Jan 2005, from MaxSonar EZ-1 Data Sheet: http://www.cytron.com.my/datasheet/sensor/LVEZ1.pdf

[5] Muhammad Ali Mazidi, Rolin McKinlay, Danny Causey (2007). PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18 (2th ed.). DeVry University: Prentice Hall.

[6] McKenna, J.C.; Anhalt, D.J.; Bronson, F.M.; Brown, H.B.; Schwerin, M.; Shammas, E.; Choset, H.; , "Toroidal skin drive for snake robot locomotion," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* , vol., no., pp.1150-1155, 19-23 May 2008

[7] Kai-Hsiang Chang; Yung-Yaw Chen; , " Efficiency on Snake Robot Locomotion with Constant and Variable Bending Angles" *IEEE International Conference on Advanced Robotics and its Social Impacts*, vol., no., pp., 23-25 Aug. 2008