Proceedings of the
International Conference on Mechanical Engineering 2011
(ICME2011) 18-20 December 2011, Dhaka, Bangladesh

ICME 11-RT-037

A BINARY HYPER REDUNDANT ELEPHANT TRUNK LIKE ROBOT CONTROLLED BY MICROCONTROLLER AND PLC-WINCC

M. Hasan[1], **R. Bunchan**[1], **I. E. L. Bohez**[2], **M. A. Wakil**[3], **M. A. Islam**[3] **and M. M. Rahman**[4]

[1]Master Student, Industrial System Engineering, School of Engineering and Technology, AIT, Thailand,
[2]Associate professor, Industrial System Engineering, School of Engineering and Technology, AIT, Thailand
[3]Assistant professor, Department of ME, KUET, Khulnla, Bangladesh
[4]Lecturer, Department of ME, RUET, Rajshahi, Bangladesh

ABSTRACT

The word hyper-redundant refers to the robot manipulators that have a large or infinite degrees of kinematic redundancy. These robots resemble in shape and operation to snakes, elephant trunks, or tentacles and thus termed bio-inspired. Binary actuation, here in this study pneumatic cylinder, was the mechanical analogy to digital electronics, where actuators flip between two discrete states on and off. For applications where high repeatability, low cost, no necessity for feedback control and reasonable accuracy are required, hyper-redundant binary manipulators based on parallel platform are potential candidates. Here, in this study two different algorithms have been applied to control the robot, a 5-module binary robot built with the aim of demonstrating the potentials of hyper-redundant binary manipulator based on parallel mechanism. The Microcontroller, based on Genetic Algorithm (GA for solving inverse kinematics problem of the binary mechanism) along with a connecting board to the solenoid valves, was found to be a successful controller of the Elephant Trunk like Robot with some ineluctable drawbacks analyzed in detail. Afterword PLC controller was implemented where WINCC acted as the Human Machine Interface (HMI) and considered as a primary solver of the limitations appeared by the Microcontroller with remaining few constraints itself too. A 3-D model of the robot has been drawn in Solid Work and analyzed the motion of the robot with the help of COSMOS Motion Analysis in Solid Work 2010. The program for controlling 30 solenoid valves of 5 module robot (each module 6 cylinders) with 30 inputs and 30 outputs was written in PLC. Visual graphics for both manual and automatic mode control were drawn in WINCC. And finally a substantial comparison between two controllers was made.

Keywords: Microcontroller, Plc, Wincc, Genetic Algorithm, Hmi, Solid Work.

1. INTRODUCTION

Typically, in industrial tasks conventional manipulators which have continuous-range-of-motion actuators and relatively low number of degrees of freedom are used. The tasks that require the manipulators to move in simple, obstacle free environments or where the manipulation of an object can be done via an end-effector, hyper redundant manipulators are very well suited. In future applications, robots are supposed to perform complex tasks in difficult to access zone, cluttered environments or in a condition having void end-effectors, in such circumstances conventional manipulator with its small number of degrees of freedom will find difficulty to perform adequately due to its lack of maneuverability. Therefore, it is desired to have a paradigm of manipulators with a high degree of freedom or maneuverability. This kind of manipulators is termed as "hyper-redundant manipulator" first time by Chirikjian and Burdick in [1]. The concept of manipulators with a high degrees of freedom combining with biological inspiration has led to particular hyper-redundant designs which have previously been referred to as: "tentacle", "snake-like", "octopus", "spine" and "elephant trunks" [2].

Hyper-redundant manipulation promises a number of potential applications [2, 3]. However, with the common type of manipulators actuated with continuous-range-of-motion actuators such as motor it is difficult to implement such kind of hyper-redundant manipulators, since they require sophisticated and expensive control and feedback systems to function with high accuracy and repeatability. For this reason, many researchers have studied and proposed several new reduced complexity systems, [4, 5, 6]. Among these systems, binary hyper-redundant robotic manipulators are potential candidates to be used in application where high repeatability and reasonable accuracy are required. Generally, binary hyper- redundant manipulators is actuated by a number of binary actuators, which have only two stable states. As the result, it is relatively inexpensive, lightweight, has a high payload to arm weight ratio, and no feedback is required. The higher number of binary degree of freedom in the system increases the capabilities of the device much better than

that of a conventional continuous robotic system. And in principle, one can make a comparison to see the analogy between continuous vs. binary manipulators and analog computer vs. digital computer [7, 8, 9].

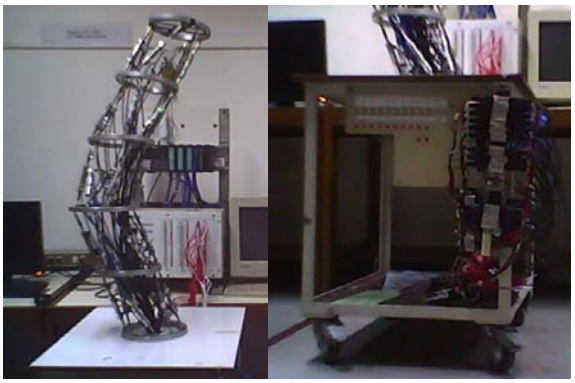
Although many potential applications of binary hyper-redundant manipulators have been shown by a number of researchers [7, 9], to date, binary hyper-redundant manipulators have remained largely laboratory curiosity. The main reasons lying behind include the complexity of inverse kinematics analysis and the suitableness of control algorithm.

The earliest studied method, also the simplest method, is brute force search, which involves searching through a discrete set of all configurations to find out the one that best matches the desired state. This algorithm becomes impractical when the number of binary manipulators is considerable large. Gregory S. Chirikjiian, Imme Ebert-Uphoff and David S. Lees proposed a great deal of theory that has laid a foundation for the inverse kinematics of binary hyper-redundant manipulators. Their work included the method based on back-bone curve [7], combinatorial approach [10] and the method using workspace density concept [11, 12]. In addition, Dubowsky S. [13] has made a comparison between the combinatorial search method and genetic search method, which showed that with large number of actuated degrees of freedom (more than 40), genetic search algorithm represents a higher performance with respect to the other.

The controllers have been used previously include PID, Fuzzy logic, Microcontroller and so on. Here in this study 'Microcontroller AT89C51RD2' has been used to control a 5 module prototype, as well as PLC (Programmable Logic Controllers) has also been used in to control the same, where WINCC is used as HMI (Human Machine Interface). In both cases of two different controllers, the inverse kinematics was solved by Genetic Algorithm (GA).

2. PROTOTYPE DEPICTION

The concept of Stewart Platform (SP) parallel binary actuation using pneumatic power supplies and air cylinder actuators are low cost, low weight of structure and can move in three dimension of workspace. For demonstration, the prototype of robot was built (in the previous work) by using 5 SP modules attached on the top of each other with 30 pneumatic cylinders in total, as shown in Fig. 1. Each module belongs a parallel SP mechanism which has 6 degrees of freedom and composes of 6 cylinders; tow plates (the base and the moving platform) and 12 ball joints. 3D model of SP mechanism was created in Solid Work for demonstrating the mechanism and physical shape. The kinematic behavior and workspace analysis of the manipulator was also explained extensively in previous work. Here in this study we are concerned to the control algorithm of the manipulator and the solution of inverse kinematics. Two different types of control algorithm were implemented and the corresponding results are discussed in detail. The program for solution of inverse kinematics problem is written in Genetic Algorithm (GA). The comparison among two controllers was also made in this study.



(a) Top (b) Bottom

Fig. 1. 5 modules, 30 cylinders Elephant Trunk like binary SP manipulator.

3. INVERSE KINEMATICS GA SOLUTION

Genetic Algorithm for solving Inverse Kinematics of n-module binary robot is presented in this section excluding the solution of Forward Kinematics as FK is quite straight forward and simple. In the process of a GA, first, an initial population of choromosomes for the GA is generated, usually in a random way. Then, the value of a function called fitness function is evaluated for each chromosome of the population. After this, the genetic operators, reproduction, crossover and mutation are used in succession of evaluation and creation of new successive generations until the satisfaction of a convenient termination condition. The basic components of GA for solving the problem are discussed below.

3.1 The input and output of the algorithm

Input: is the desired position which the end-effector of the manipulator needs to reach. This desired position is represented as a vector of 6 parameters, three translation parameters and three rotation parameters $q = [x_d\ y_d\ z_d\ \alpha_d\ \beta_d\ \gamma_d]$.

Output: is the configuration of the manipulators i.e ON/OFF state of the cylinders with which the distance between desired position and calculated position lies in the range of a predefined threshold. Simply, the output indicates which binary actuator should be 'On' and which one is 'Off' so that the end-effector of the whole manipulator is close to the desired position within a small error.

3.2 The evaluation mechanism: Fitness function

This mechanism consists of evaluating a function called fitness function for each chromosome of the GA's population. Denote $q_d = \{ x_d\ y_d\ z_d\ \alpha_d\ \beta_d\ \gamma_d \}$ as the 6 coordinates (both translation and rotational parameters) of the end-effector relative to the base (0,0,0). The fitness function of one chromosome in GA is determined as follows:

- Use the chromosome (manipulator state or configuration) as the input of Forward Kinematics algorithm to find out the 6 coordinates of end-effector $q = \{x_d\ y_d\ z_d\ \alpha_d\ \beta_d\ \gamma_d\}$, not shown in this paper.
- Fitness function
  $Fitness = 1/(W_t \bullet P_{error} + W_r \bullet O_{error})$
  Where, P_{error} is positional error,

$P_{error}=\sqrt{(x-x_d)^2+(y-y_d)^2+(z-z_d)^2}$

O_{error} is orientation error,

$O_{error}=\sqrt{\Delta\alpha^2+\Delta\beta^2+\Delta\gamma^2}$

$\Delta\alpha=(\alpha-\alpha_d)$; $\Delta\beta=(\beta-\beta_d)$; $\Delta\gamma=(\gamma-\gamma_d)$

W_t and W_r are translational and rotational weight factors respectively

In fact, each orientation error is normalized on the range of $(-\pi, +\pi)$. The algorithm for normalization is described below.

Input: Two angles $\theta_1$and θ_2

Output: Angles difference $\Delta\theta \in (-\pi, +\pi)$

$\Delta\theta = \theta_1 - \theta_2$

if $(\Delta\theta < -\pi)$ then $\Delta\theta = \Delta\theta + 2\pi$

else if $(\Delta\theta < \pi)$ then $\Delta\theta = \Delta\theta - 2\pi$

return $\Delta\theta$

The purpose of the GA is to maximize the fitness' value or minimize the error given by P_{error} and O_{error}. This fitness function will guarantee the survival of 'good' chromosomes in the population which represent feasible solutions to the real problems.

3.3 GA parameters

Reproduction: natural selection of survival-of-the-fittest

Crossover: recombination of chromosomes

Mutation: randomly alters the value of each gene.

Population size: module dependent

Crossover rate: chosen around 60 – 70 %, after many trial and error tests

Mutation rate: chosen around 1– 3 %, after many trial and errors tests

Scaling: avoids the problem of premature convergence in GA. In this work, a sigma-truncation technique is applied.

Fitness = Fitness $_{raw}$ – (Fitness $_{average}$ – Scaling Factor, σ)

Where Scaling Factor a small integer chosen in the range [1- 5] and σ the population's standard deviation

Termination: The algorithm is defined to terminate when one of the two following conditions occurs

1. The position error produced so far is less than an acceptable minimum threshold. This threshold value is set depending on the given input position. It is generated by using approximate positional accuracy algorithm.
2. The number of current generation is greater than the predefined value. The predefined number of generation is set depending on the number of binary actuators, also. It is a large number if the number of bits is increased and vice versa.

3.4 GA Results

First GA result is mentioned in Table.1, with the following parameters excluding rotational parameters (roll, pitch, yaw) for simplicity.

Parameters:

Position_Weight $W_t = 1.0$; Rotation_Weight $W_r = 0.0$

GA parameters:

Number of modules: 5

Number of individuals: 200

Number of generations: 700

Mutation rate: 0.02; Crossover rate: o.6

Scaling: 1

```
Procedure GA:
Begin
  Check far valid input using workspace-
      approximation algorithm
  First Generation
  Initialize Population (Random)
  Evaluate Population
      Using FK algorithm to find out {x, y, z, α,β,γ}
      Fitness = 1/ (Wt • Perror + Wr • Oerror)
  Repeat
      Generation⟵ Generation + 1
      Select Population (Generation) –
          from Population ( Generation -1)
      Gen Operators
          Crossover
          Mutating
          Scaling
      Evaluate Population
          Using FK algorithm to find out  {x, y, z, α,β,γ}
          Fitness = 1/ (Wt • Perror + Wr • Oerror)
      Population (Generation -)⟵Population (Generation)
  Until (Termination Conditions)
```

Fig. 2. GA program.

Total error (distance between the given position and the solution):

$Error=W_t \bullet P_{error}+W_r \bullet O_{error}=\sqrt{(x-x_d)^2+(y-y_d)^2+(z-z_d)^2}$ (mm)

Table.1. shows the result of GA solution in the case of testing with 5-module binary manipulator without rotation parameters (roll, pitch, and yaw).

Table 1: GA solution for 5-module manipulators without rotational parameters

Desired position *(x_d, y_d, z_d, $roll_d$, $pitch_d$, yaw_d)*	GA solution *(x, y, z, roll, pitch, yaw)* corresponding manipulator state	Running time (s)	Total error (mm)	Positional accuracy (mm)
(-9.4, 12.9, 1215.7, 0, 0, 0)	(-9.313, 13.078, 12115.432, 1.040, 0.079, 0.009) 00010110010110001010000010000010	27.265	0.340	0.18
(150, 150, 1200,)	(150.653, 149.686, 1200.152, 1.286, 0.140, 0.195) 01000010111110010110100000001	26.828	0.740	0.226
(10,10,1300, 0, 0, 0)	(10.805, 10.747, 1299.360, 0.812, -0.164, -0.069) 11101011001111111111010110111	17.449	1.271	1.797
(200, 200, 1200, 0, 0, 0)	(200.575, 199.786, 1200.334, 2.430, -0.097, 0.387) 01010100011110011110110010001	27.953	0.698	0.283
(0.06, 0.12, 1363.27, 0, 0, 0)	(0, 0, 1363.828, 1.047, 0, 0) 11111111111111111111111111111	6.584	0.573	0.0 (2.0)
(0.02, -0.28, 1209.34, 0, 0, 0)	(0.0004, 0.0002, 1209.78, 1.047, 1.054, 0.159, 0.092) 10010001101110010001010110101 0	29.382	0.523	0.199
(59.01, 34.2, 1343.5, 0, 0, 0)	(58.998, 34.062, 1343.76, 1.054, 0.159, 0.092) 11111111111111111111111111100	17.906	0.296	0.0 (2.0)
(-0.21, 0.05, 1302.2, 0, 0, 0)	(0, 0, 1302.112, 1.047, 0, 0) 00000011111111111110000000111111	20.062	0.232	1.701
(-653.05, -276.94, 976.16, 0, 0, 0)	(-653.174, -377.108, 976.421, 1.193, -0.634, -0.438) 11000011110000110000111101 1011	2.750	0.335	0.0 (2.0)
(500, 156, 1350, 0, 0, 0)	Warning: out of workspace			0.0

4. CONTROL ARCHETECTURES

The methods for controlling the binary robot are presented in this section. Two methods have been used for controlling the robot. WIN CC software has been used for Human Machine Interface. The user has to input the desired positions; GA will solve the corresponding configurations (on/off) of pneumatic cylinders. The controller will follow the configurations found in HMI and allow solenoid valves to open or close and thus operating the cylinders, referring the desired movement of robot. The methods of control algorithm are discussed in subseeding paragraphs.

4.1 Microcontroller

Fig.2 shows the flow diagram of the binary robot control starting from the user interface in PC; the solution is calculated for an input position using GA, then the obtained manipulator state is sent to microcontroller through Serial RS232 port of PC as a string of bits. After processing the string bits received, the controller sends signal to the Solenoid controller to activate solenoid valves. Solenoid valves allow compressed air to go into cylinders and thus actuate.

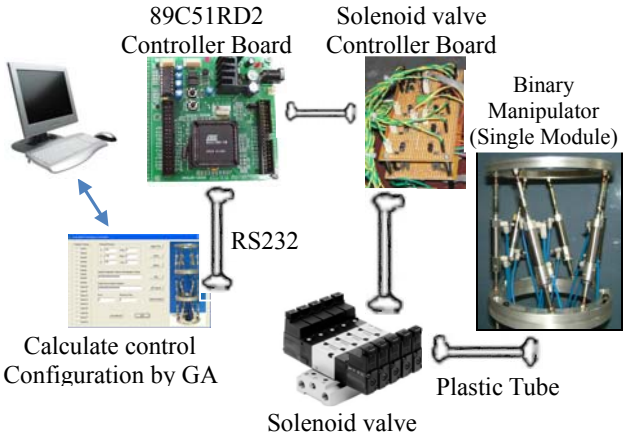


Fig. 2. Flow diagram of binary robot control using microcontroller

Users need to enter the desired position (x, y, z, roll pitch, yaw). For example (10, 10, 1300, 0, 0, 0), after insertion of values GA calculates out the solution for corresponding manipulator configuration which will be shown in HMI in terms of bits for example the solution for above end position is 11101011001111111111110110110111. The manipulator configuration will then be sent to microcontroller. Microcontroller through solenoid controller board sends command to solenoid valves. And solenoid valves operate pneumatic cylinders according the received directions. HMI also shows the status of the manipulator which cylinders are on and which is off.

4.2 Programmable Logic Controller (PLC)

The layout of system flow is shown in Fig. 3, where PLC is replaced by removing existing Microcontroller and Solenoid valve controller board seen in Fig.2. The desired configuration of the manipulator is presented in HMI with the help of WINCC. Then corresponding commands go to PC and then to PLC through MPI cable. The PLC program is written in Ladder Logic language by virtue of Simatic step 7. PLC is connected to the Solenoid valve and sends the direction according to the configuration given by users. And finally solenoid valves operate the actuators i.e cylinders as like in case of microcontroller.

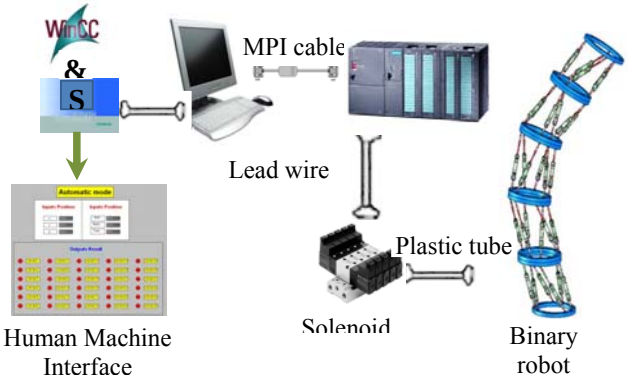
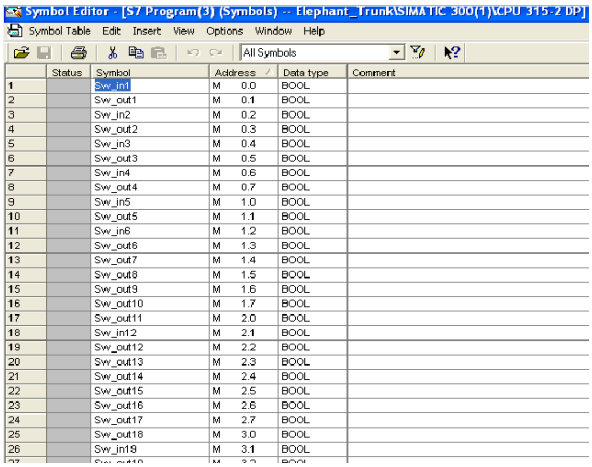


Fig. 3. Control schematics of robot using PLC

4.2.1 Number of symbols and I/Os

There are 49 solenoid valves available with the robot, to control 30 cylinders of the manipulator. Out of 49, 38 solenoid valves are 5/2 double solenoid valves and the rest 11 are 5/2 single solenoid valves. For each of double solenoid valves two symbols are necessary as example 'Sw_in1' to indicate solenoid valve 1 is 'in' and 'Sw_out1' to indicate solenoid valve 1 is 'out', whereas 'in' means air-in in the solenoid and 'out' means air-out from the solenoid. Likewise, for each single solenoid valve only one symbol is enough to indicate its action as example 'Sw_in1' to indicate solenoid valve 1 is in and it comes out by spring set in the valve. Therefore, the total number of symbols referred to the PLC is 38 multiplied by two and sum with 11 which equals 87 symbols. To control 30 cylinders, there are 30 inputs interact to the plc and 30 outputs from the same. The symbols indicating the input and output status with their addresses in the CPU are shown in the Fig. 4.



	Status	Symbol	Address	Data type	Comment
1		Sw_in1	M 0.0	BOOL	
2		Sw_out1	M 0.1	BOOL	
3		Sw_in2	M 0.2	BOOL	
4		Sw_out2	M 0.3	BOOL	
5		Sw_in3	M 0.4	BOOL	
6		Sw_out3	M 0.5	BOOL	
7		Sw_in4	M 0.6	BOOL	
8		Sw_out4	M 0.7	BOOL	
9		Sw_in5	M 1.0	BOOL	
10		Sw_out5	M 1.1	BOOL	
11		Sw_in6	M 1.2	BOOL	
12		Sw_out6	M 1.3	BOOL	
13		Sw_out7	M 1.4	BOOL	
14		Sw_out8	M 1.5	BOOL	
15		Sw_out9	M 1.6	BOOL	
16		Sw_out10	M 1.7	BOOL	
17		Sw_out11	M 2.0	BOOL	
18		Sw_in12	M 2.1	BOOL	
19		Sw_out12	M 2.2	BOOL	
20		Sw_out13	M 2.3	BOOL	
21		Sw_out14	M 2.4	BOOL	
22		Sw_out15	M 2.5	BOOL	
23		Sw_out16	M 2.6	BOOL	
24		Sw_out17	M 2.7	BOOL	
25		Sw_out18	M 3.0	BOOL	
26		Sw_in19	M 3.1	BOOL	

Fig .4. I/O symbols with their addresses.

4.2.2 PLC program (Ladder Logic)

The PLC program is written in Ladder Logic language with the help of Simatic Step 7 program. As is observed in Fig. 5, two normally closed functions have been used in Network 1 and one normally open and one normally closed function is used in Network two. Whenever no switch is turned on the signal goes through normally closed function and the cylinder is 'in' condition. When Sw_in1 is turned on Network 1 is off mode as it opens the circuit by means of normally closed function as well as it closes Network two with the help of normally open function resulting the cylinder 1 is 'out' condition.

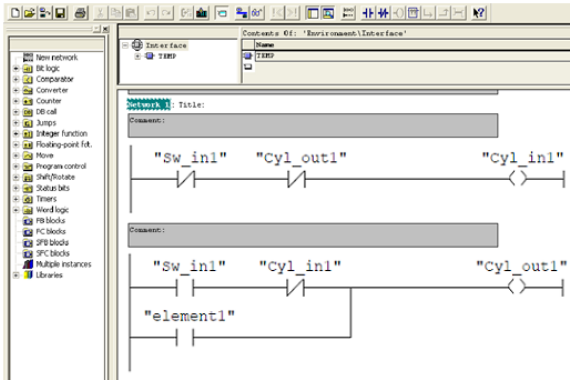


Fig 5. PLC Ladder Logic diagrams

5. RESULTS AND DISCUSSIONS

The results obtained with the controller of PLC are reported by the graphical representation of HMI**.** Simatic WinCC has been used as HMI in this study as mentioned earlier. Fig. 6 shows the graphical representation of Human Machine Interface. Two functions have been utilized primarily which describes the status of the plant and type of mode as shown in figure. Simply plant status can be either on or off and there have been used two modes in this study to control the robot, manual and automatic.

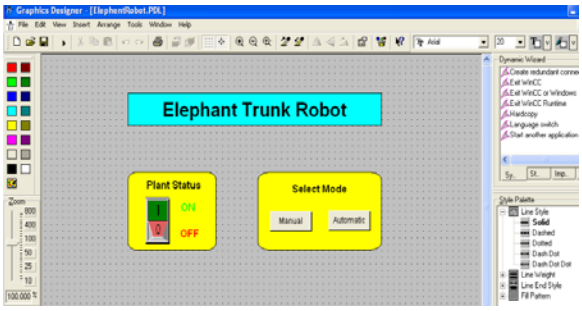


Fig.6. Graphical representation of HMI

5.1 Manual mode

30 buttons have been created for each of 30 cylinders as mentioned in Fig. 7. The bottom position of the on-off button indicates that the cylinder is in position and vice versa. Whenever, as example, cylinder 1 button is turned on the actual pneumatic cylinder is goes out and vice versa. Thus the Manual mode of the HMI operates.

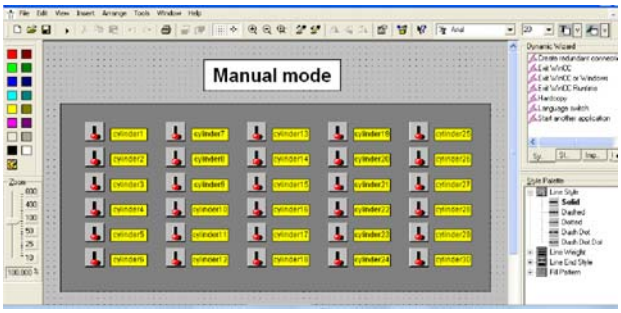


Fig 7. HMI manual mode

5.2 Automatic mode

The Automatic mode graphical representation of the HMI in WinCC is mentioned in Fig. 8. The value of X, Y, Z and roll pitch yaw is put in input position box. After deriving the calculation for the configuration of the manipulator, the corresponding cylinder is turned on and thus operates the robot.

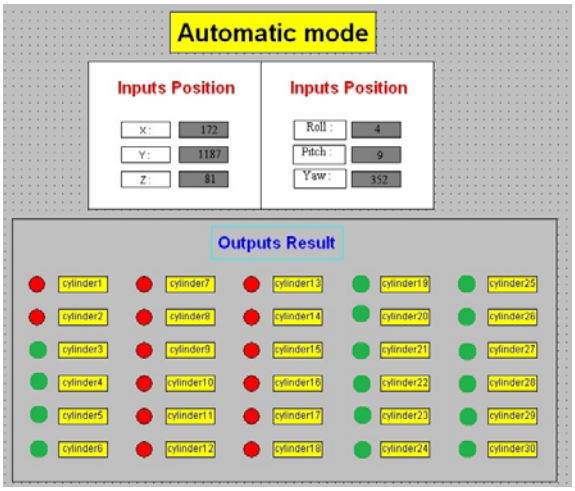


Fig 8. HMI Automatic mode

6. CONCLUSIONS AND RECOMMENDATIONS

The Control performance of the Hyper Redundant Binary Bio-Inspired Elephant Trunk Robot using PLC and WinCC as HMI is better from the view points of fastness, accuracy and robustness than microcontroller. However, a comparison of using PLC Microcontroller has been made in the succeeding paragraph following the advantages of using Binary robot and few recommendations for future work.

6.1 Comparison: PLC vs Microcontroller

- PLCs are easily programmed using Ladder Logic, Function Blocks, or even Statement List than fabricating microcontroller along with the solenoid controller board.
- PLC running time was much shorter than microcontroller and thus it is found faster.
- The desired end-position was achieved by PLC was more accurate than microcontroller.
- Ladder Logic programs which appears very similar to industrial schematics. This allows

electrocutions quickly modify/change the program. Micro controller has a very different programming language, assembly, basic, etc.

- Usually a PLC is used in an industrial environment. Where a micro-controller is smaller and well suited for embedded situations.
- In plc, all inputs and outputs are scanned in each cycle and each part of program is executed separately and simultaneously but in microcontroller the program run from first line to end when the program, for example, in line10 it cannot see an input that use, for example, in line20.
- PLC voltage range of input and output are higher than Microcontroller. It suitable to use with relay that has higher coil voltage usually 24VDC and sensor or components that work with higher voltages (usually 24VDC).
- PLC facilitates an easily changeable operating system as well as user program.
- For miniaturization, Microcontroller possesses promising advantages over PLC.

6.2 Advantages of Binary robot

Binary robots, as example used in this study, have different attractive advantages. Some of them are mentioned as follows.

i. Discrete states and high repeatability
ii. Relatively low cost and light weight
iii. High capacity load to manipulator weight ratio
iv. No need for feedback control
v. Less complexity in computer controlled interfacing
vi. Allow tasks to be performed even when some actuators fail
vii. High ability in obstacle avoidance.

6.3 Recommendation

The presented mechanism if attached to applications like pick and place or path planning, will refer the real application of such kind of mechanism. It would be a good approach to combine binary robot with continuous robot. For example, the proposed manipulator will become very flexible if it includes 6 modules: 5 binary Stewart Platform modules and one continuous module at top. Using this combination, the propose manipulator will be able to move exactly to a give position in the workspace. However, the forward kinematics and inverse kinematics problem will need more efforts to be solved and the controller is also more complicated.

Considering the problem of building a binary robot with a considerable number of binary actuator, the development of the new kind of binary actuators, which follows the requirements on lightweight, low cost and ease of controlling, has become great interest. With these new kinds of binary actuators, the binary mechanisms composing a large number of modules become more practical. The alternative installation of binary robot should be considered, including the effects on different installation such as over the ceiling or on the wall

6. REFERENCES

1. Chirikjian, G.S., and Burdick, J.W., May, 1990, 'An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulator'. Proc. Of IEEE ICRA'90, Cincinnati, OH.
2. Chirikjian, G.S., and Burdick, J.W., December, 1994. 'Hyper-Redundant Manipulator'. IEEE Robotics and Automation Magazine, pp. 22-29.
3. I.D. Walker, August, 2000. 'Some Issues in Creating 'Invertebrate' Robot'. Proceedings of the International Symposium on Adapting Motion of Animals and Machines, Montreal, Canada.
4. Canny, J., Goldberg, K., 1993. 'A RISC Paradigm for Industrial Robotics'. Tech. Rep. ESRC 93-4/RAMP 93-2, Engineering and Robotic System, Vol. 19, page 5-12.
5. Craig, John J. 'Introduction to robotics: mechanics and control'. Library of Congress Cataloging-in-Publication Data 1955.
6. Goldberg K., 1992. 'Orienting polygonal parts without sensors. Algorithmic, Special Robotics Issue'.
7. Chirikjian, G.S., 1977. 'Inverse Kinematics of Binary Manipulators Using a Continuum Model'. Journal of Intelligent and Robotic Systems, Vol. 19, pages 5-12.
8. Suthakorn, J., Chirikjian, G.S., December, 2000. 'Design and Implementation of a New Discretely-Actuated Manipulator'. International Symposium on Experimental Robotics (ISER), Hawaii.
9. Chirikjian, G.S., 1994. 'A Binary Paradigm for Robotic Manipulators'. Proceeding of the 1994 IEEE International Conference on Robotics and Automation, pp. 3063-3069.
10. Kittipong C., Thesis (M.Eng.) - Asian Institute of Technology, 2003. Development of an elephant trunk-like robot actuated by parallel binary manipulators, Thailand.
11. Lees, D. S., Chirikjian, G.S., 1996.'A Combinatorial Approach to Trajectory Planning for Binary Manipulators'. Proc. IEEE Int. Conf. on Robotics and Automation, Minneapolis, MN, pp. 2749-2754.
12. Jean-Pierre Merlet, INRIA Shophia-Antipolis, 2000. 'Parallel Robots. Kluwer Academic Publishers'.
13. Dubowsky, S., Lichter, M.D., Sujan, V.A., May, 2002. 'Computational Issues in the Planning and Kinematics of Binary Robots'. Proceedings of the 2002 IEEE International Conference on Robotics and Automation. Washington, D.C.