REHAB *Tech*

Monash Rehabilitation Technology Research Unit

The Sir -1 Robot

Steven Thomson

MONASH

AUSTRALIA'S INTERNATIONAL UNIVERSITY

REHAB *Tech*- Monash Rehabilitation Technology Research Unit assume no liability for any claim of adverse effects resulting from misapplication of the information presented here in. While every effort is made to ensure the accuracy of the guide no responsibility or liability will be taken for any inaccuracies.

REHAB*Tech* is finance and supported by

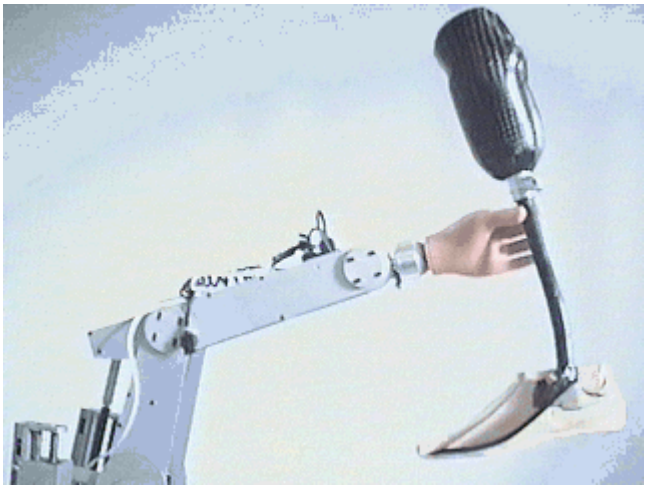
Commonwealth Department of Veterans' Affairs

In collaboration with

MONASH
AUSTRALIA'S INTERNATIONAL UNIVERSITY

REHAB *Tech*- Monash Rehabilitation Technology Research Unit
C/- C.G.M.C.
260 - 294 Kooyong Road
CAULFIELD VIC 3162
AUSTRALIA
Email rehab.tech@eng.monash.edu.au

A Computerised Robot Arm with a computer so that the combination can be used in upper limb functional and training assessments and as a hands- on display item. The Sir - 1 is a six axes robot modelled similar to the human arm. The robot has a shoulder, elbow, wrist (with control of pitch and roll), gripper and a base. Sir- 1 will be used in the assessment of upper limb control through the use of various attachments that will plug into the IBM computer that is connected to it. The user shall be able to control the robot to varying degrees which will allow us to isolate specific control actions.



The Sir-1 robot came to **REHAB*Tech*** courtesy of Baldwyn Petrochemicals. It was in less than perfect condition - the shoulder system was not working and without a gripper (hand). The first step was to get the robot up and running and then to find a suitable gripper. Three grippers were evaluated - the SIR-1 gripper which was leant to **REHAB*Tech*** by Ron Viney of the Metropolitan College of TAFE, an Otto Bock Greifer and an Otto Bock electric hand. The evaluation resulted in the greifer being selected due to its superior reach, grip and speed. The gripper was too slow and small while the Otto Bock electric hand an inferior grip to the greifer due to the circular grip of the fingers.

A number of different programming methods may be used to control the robot - the teach pendent, host mode and programming through the computer. The teach pendent is directly connected to the SIR- 1 control and any program entered there is saved in the control's memory, host mode relies on a menu written in basic which allows you to use most of the commands of the teach pendent from the computer keyboard and then there is programming the robot from basic which gives the most freedom and allows for other equipment to be used to control the robot such as joysticks, EMG sensors, key pads and switches.

Presently, the robot has been repaired and improved through the use of the greifer, its programming has been researched, various methods have been put into use and assessed and, the robot has been successfully connected to an IBM PC. In the future, some input equipment will be attached and the whole system will be ready for trial

Abstract

The SIR- 1 is an intelligent multipurpose six axes robot manufactured by Scien - tech Intraco Automation. The basic system configuration of SIR - 1 consists of a robot arm, a micro processor based controller and a hand held teach pendent.

Using a computer interface the aim was to enable SIR - 1 to be operated through control systems used by amputees. The system then could be used as a functional learning tool for patients during early use of prosthesis.

Introduction

Controller

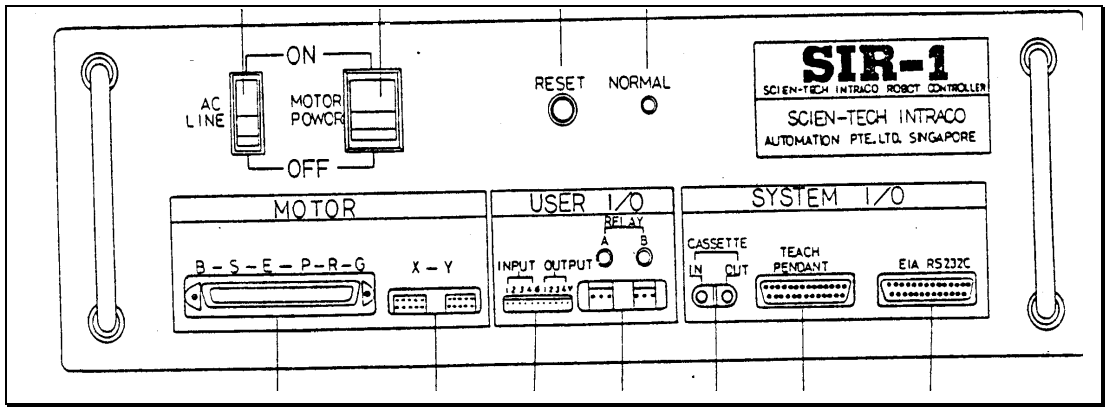
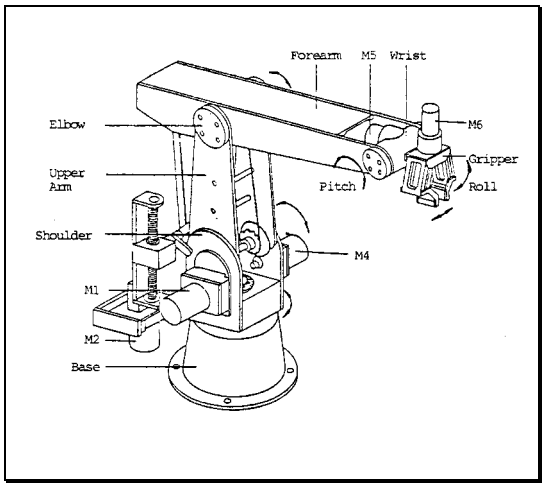


fig 1 *frontal view of the SIR- 1 Controller*.

◊ **MAIN FIGURES**

Input / output ports	=> providing synchronisation with other devices.
Relay Ports	=> providing synchronisation with other devices, LED lit when relay activated.
Teach Pendent	=> connection for hand held programmable pendant.
RS - 232 Serial Interface Port	=> providing connection to computer with standard RS - 232 serial interface
Robot Arm Port	=> connection operating robot arm.
Accessory Port	=>providing connection of accessories to operate simultaneously with robot arm.



SIR - 1 has six axes of movement. Base, Shoulder, Pitch, Elbow, Roll and Gripper. The gripper shown in diagram was replaced by an Otto Bock hand operating through the relay ports mentioned above.

fig 2 *SIR - 1 Robot*

Procedure

◊ **SET UP**

Previously SIR - 1 had been interfaced with a 486 computer. The SIR -1 controller RS232 port was altered as shown to communicate with the PC. Using Qbasic programming language simple programs were written configuring the PC com port with the RS232 port of SIR - 1. The baud rate was set at 9600 bps with a eight bit word length, 1 parity bit and no stop bit as advised in the SIR - 1 documentation.

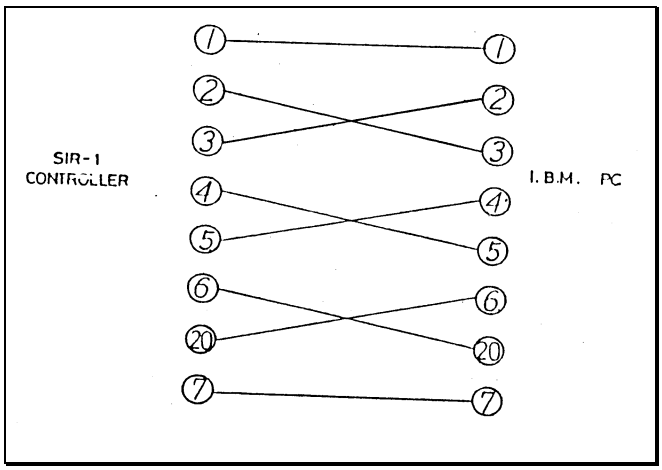


fig 3 *RS232 Port interface with SIR - 1*

◊ **PROGRAMMING**

1. Programs were written to make SIR - 1 carry out specific tasks
   - writing sentences
   - picking up glass / pouring

2. A program was written enabling the robot to be operated via keyboard function keys.
   [see appendix 1]
3. A program was written enabling the robot to be operated through the input pins of the I / O port.
   [see appendix 1]
4. An addition to the above program enabled the position of each axes to be viewed on screen.
   [see appendix 1]

5. A basic switching circuit was developed using relays, enabling input pins to be activated by Otto Bock digital electrodes.

Methods

1. Specific Tasks

The aim of this procedure was to become familiar with the set up and to use SIR - 1 to simulate human movement. This was done by programming sets of instructions involving basic movements and pauses of the 6 motors. fig shows an example program

```
Open "COM 2:9600,N,8,1" For Random As #1
Print #1, CHR$(27)
Print #1,"M0,30,40,0,0 ,0,0,0,32"
Print #1,"P 20"
Print #1, "M0,-30,-40,0, 0,0,0,0,16"
```

Line 1- configures the port for communication with SIR - 1 Controller.
Line 2 - sends handshake signal to initialise communication.
Line 3 - 5 - Move and pause commands.
Each value in the print statement operates a different motor. The last value is used to activate relay to open and close gripper.

32 opens hand
16 closes hand

fig 4 *Programming basic movements*

Programs worked well however lack of synchronisation meant longer programs tended to get jumbled and crash.

2. *Keyboard Function Keys*

The basic movement commands shown in above example were used as commands in subroutines activated by function keys. Each function key activated a different subroutine which in turn moved a specific axes in a specific direction. [see appendix]

3. *Input / Output Port*

It was found that using a read statement the I/O port state gave a value between 1 and 15. As input pins were tied high when inactive. This number was the binary product of the four input pins.

fig 5 *shows translation*.

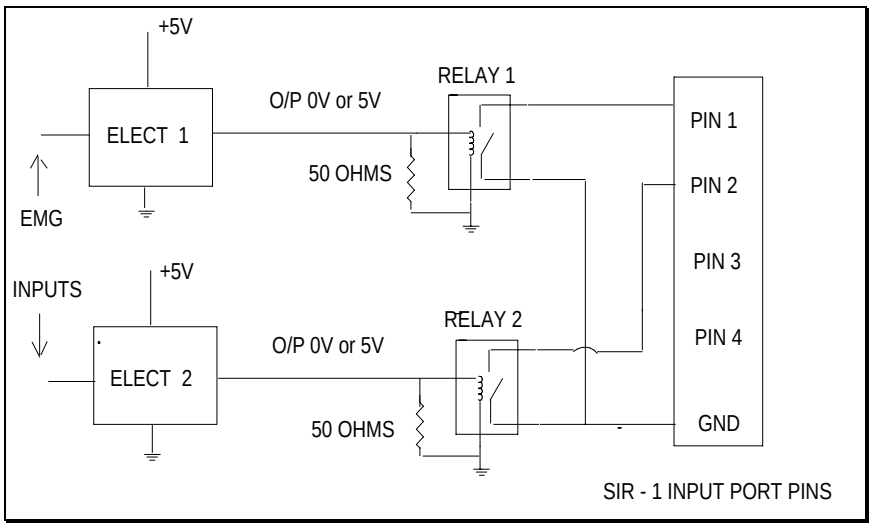
PIN 1	PIN 2	PIN 3	PIN 4	Computer	Translation
1	1	1	1	15	No pins active
0	1	1	1	7	Pin 1 active
1	0	1	1	11	Pin 2 active
1	1	0	1	13	Pin 3 active
1	1	1	0	14	Pin 4 active

This enabled a specific move command to be set to a specific input pin, so when a pin was activated the robot would move a certain distance in a set direction.

4. *Position Display on Screen*
The Read statement also allowed the position of each axes to be continuously updated on the computer screen for reference.

5. Otto Bock Digital EMG Electrodes
Enclosed Differential Filter circuit with +5V supply designed to generates a +5V signal when activated my an EMG signal produced by stimulated muscle.

6. *Relay Switching Circuit*



The idea of this circuit was to interface the electrodes with the input pins to operate the robot.

◊ The Circuit
⇒ EMG Signal
⇒ +5V
⇒ Relay closed
⇒ Pin brought low

fig 6 *Relay Circuit*

DISCUSSION

♦ **FUNCTIONAL ASSESSMENT TOOL**

Although SIR -1 could be operated via electrodes, the jerky motion of the robot meant that if used as a functional assessment tool, the control of the robot by the computer would be assessed rather than how the subject operated it via the electrodes.
To overcome this problem a program enabling continuous movement of the robot rather than movement in numerous set amounts has to be found.

♦ **PROBLEM**

Programs both reading from and writing to SIR - 1 Controller tend to randomly crash producing an Err 24 [Device Time out]. This means that computer is not receiving data from SIR - 1 controller in a set time.
Possible Courses

- *Interface when both reading and writing jams up.*
  ⇒ Initially thought the problem but programming wait states between the two functions which was thought would fix this problem didn't change the error.

- *Faulty Connection between machines - data lost*
- *Problem with SIR - 1 Controller*
  ⇒ Both these possibilities have not been properly investigated

SIR - 1 is still functional when operating with this error, however use can become cumbersome.