

UNIVERSITY OF CAPE TOWN

DEPARTMENT OF MECHANICAL ENGINEERING

DEVELOPMENT OF A LINEAR PUMP AND COMPUTER CONTROL INTERFACE TO SIMULATE BLOOD FLOW WITH PHYSIOLOGICAL FLOW AND PRESSURE CONDITIONS

Student: Itumeleng Pitiri (PTRITU001)

Supervisors: Assoc Prof RD Knutsen

Dr Thomas Franz (Cardiovascular Research Unit)

Mark Yeoman

Date of Submission: 28 October 2003

Project No. 55

To develop a linear-drive pump and a computer control interface which will be used in rig for testing mechanical behaviour of blood vessels.

The measurement of mechanical behaviour of natural and artificial blood vessels in laboratory test requires the simulation of flow and pressure conditions found in human body. The physiological blood pressure, blood flow rate and blood flow volume behave cyclic with the heart rate. The pressure and flow conditions also change e.g. with the degree of activity, like rest state compared to exercising. A linear-drive pump has to be developed which is capable of creating physiological pressure and flow conditions in a closed fluid circuit. The pump should be computer-controlled to allow the variation of pressure and flow profiles over the whole physiological range.

DECLARATION

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the Numbered References Convention for citation and referencing. Each significant contribution to, and quotation in, this project from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report/project is my own work.

4. I have not allowed, and will not allow anyone to copy my work with the intention of passing it off as his or her own work.

Signature.....

ACKNOWLEDGEMENTS

The author would like to thank the following people for their contribution in the project:

Above all: Mark Yeoman, Dr T. Franz and Prof. R. Knutsen for guidance and support in the project as supervisors

The author would also like to thank the following people's contribution:

- Samuel Ginsberg for helping to set up experimental equipment of the project
- Dr W. Capper from Biomedical Engineering at UCT
- Math-works support team (especially Gareth Shaw)
- Julian Mayer: For making PCI 30GA board and oscilloscope available to be used.
- Greg Perks: An engineer at FESTO Company regarding pneumatic drives.
- Software engineers at Eagle technology for assisting with software requirements for the PCI 30 board.

TERMS OF REFERENCE

This project was proposed by the Cardiovascular Research Unit / Medtronic Institute for Biomedical Research (CVRU) which is placed at the University of Cape Town Medical School, Observatory 7925, Cape Town on the 7th July 2003. The request was specifically put forward by Dr. T. Franz at the CVRU.

The proposed project is to look at the design and development of a computer-controlled pump that will eventually form part of an existing rig that is used to test mechanical properties of artificial and natural vessels.

Specific instructions put forward were to:

1. Undertake a literature review on previous work related to the thesis
2. Undertake a market research of commercially available Cardiovascular Pulse duplicators.
3. Identify an optimum design of a linear-drive pump.
4. Develop a PC interface to control the pump with suitable software.
5. To prove the functionality of the computer-controlled interface.
6. To write a report that includes all supporting documents, e.g. drawings, quotations.
7. Submit the report by 28th October 2003.

EXECUTIVE SUMMARY

This thesis concerned the design of a computer-controlled pump for testing mechanical properties of natural and artificial vessels.

The main purpose of a computer-controlled pump is to simulate heartbeats of human beings by reproducing physiologically equivalent cardiovascular conditions such as pressure and flow-rates. The pump system is to be used in the rig at the Cardiovascular Research Unit to replace an existing roller pump, as this pump is not easily controllable.

The objectives that had to be met on the thesis were:

- To determine the optimum design of a pump that can replace the roller pump. The final deliverable being detailed drawings of the pump for future manufacturing.
- To develop software program and control interface for the optimum design of a pump. The integration of the pump control to the rig shall be achieved in future but the program should be capable of generating and capturing signals. The generation of signals will ensure that the software forms input to the pump driver and capturing will ensure that the software can capture sensor signals for comparison and verification of programmed function from the rig.

The theoretical approach began with undertaking a literature review to determine requirements for suitable pumps. From a number of pumps investigated it was determined that piston pumps and diaphragm pumps could form ideal computer-controlled pumps for reproducing pulsatile physiological conditions. The main reason was that displaced fluid is directly proportional to step input of the pump i.e. stroke.

As flow-rate was found to be the controlled parameter in order to reproduce physiological conditions, proportional relationship of pump stroke and volume displaced will ensure easier control of the pump. Diaphragm pumps were chosen as the best option because of better mechanical considerations as there is no surface-to-surface contact of moving parts unlike in piston pump.

Suitable computer controlled drivers were investigated to determine best driver for the pump. The main requirement for such a system is that they must have quick response to input (e.g. electrical or pneumatic signal) to accurately reproduce physiological conditions. A pneumatic system that can control the stroke of the pump in response to an input voltage was found to be the best option. The system produces pressure signals, which in turn control the stroke of the pump in response to input voltage signals. The system consists of an electrical system for producing pressure in proportion to input voltage signal and a mechanical system for controlling pump stroke in response to pressure signal from an electrical system. It is locally available from FESTO Company. Based on specifications of the pneumatic controller, the parameters of the diaphragm pump were optimized. Detailed drawings of the pump were then produced.

MATLAB software was developed and integrated to the PCI 30GA card (A device that can be used to generate and capture signals through its channels). The developed MATLAB software program is capable of generating and capturing voltage signals. (Oscilloscope and signal generator were respectively used to test the functionality of the program). The software could generate voltage signal, which could be composed from Fourier functions by a developed MATLAB graphical user interface. However the software was only capable of capturing signals for limited frequencies and a limited time.

Solutions proposed in the project, which were a **Diaphragm pump**, **MATLAB program**, **PCI card** and a **cylinder positioner** are to be implemented in future. Their integration was not part of this thesis. However the following figure shows how they could possibly be integrated together to form part of the Compliance rig.

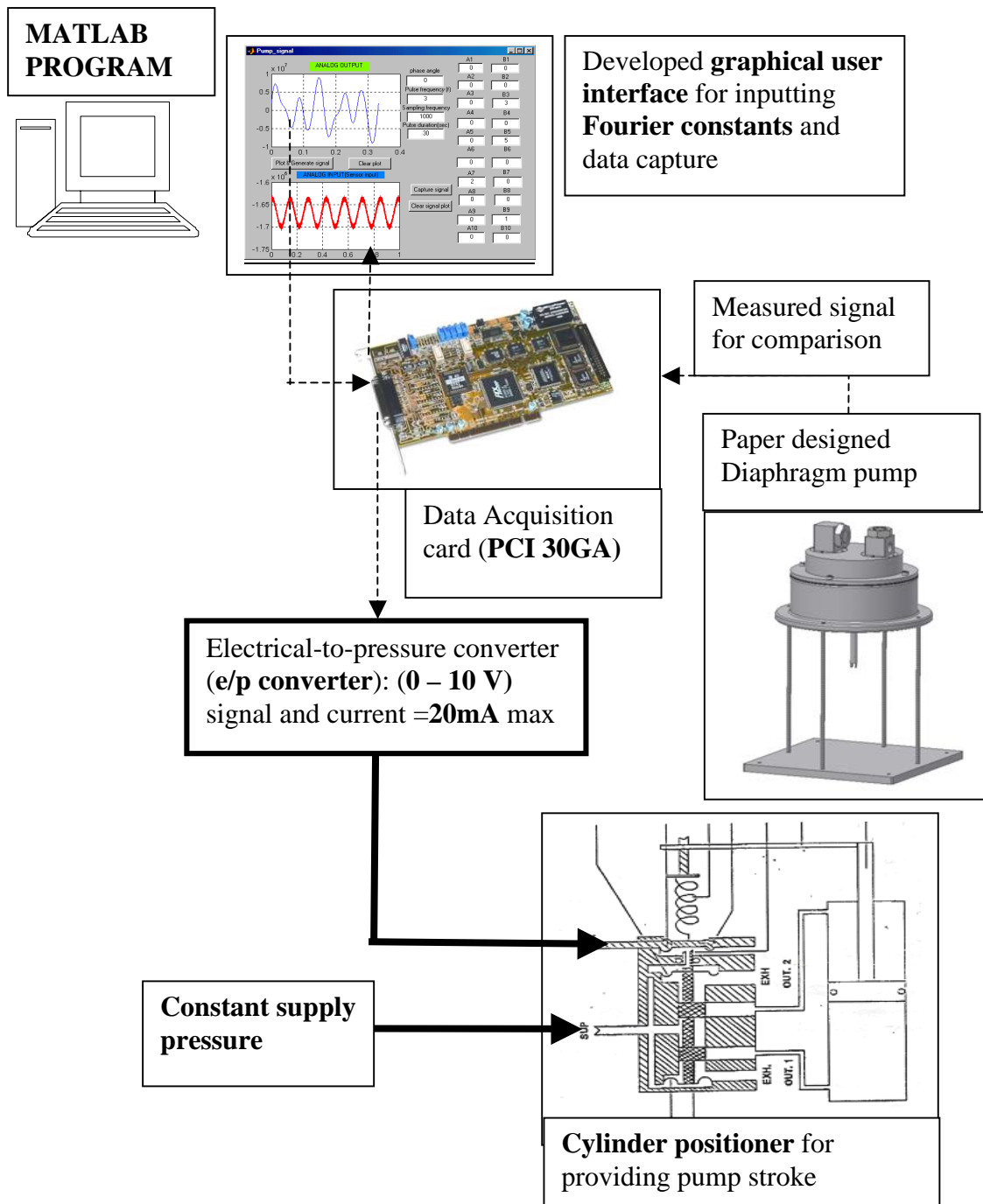


Figure showing future integration of proposed solutions

Key: ----- Voltage signals
 ————— Pressure signals

Main **conclusions** drawn are that

- Diaphragm pump can be implemented as the best options to replace the existing roller pump for better reproduction of physiological conditions. Drawings attached at the back can be used as the basis for implementing the drawing.
- The developed system can generate signals to form an input to the pump controller while it cannot accurately capture data for future use on the Compliance rig to capture data from sensor signals.

Main recommendations of the project are that:

- Pump can be manufactured once the controller system (**MATLAB program to a cylinder positioner**) that was only theoretically proven is further investigated.
- Valve design of the pump requires further investigation.
- Signal generation can be done with the developed software while data capturing can be done with freely available software *Waveview*.
- Feedback control loop on the Compliance rig must be further investigated to offer more accurate control of the pump. This would require the implementation of flow-sensors to measure how accurately programmed signals closely match physiological conditions.

GLOSSARY

Definition of terms

1. *in vivo* – Taking place inside the living organisms
2. *in vitro* – Taking place outside the living organism, i.e. in the laboratory.
3. Graft is an implant material for damaged natural or artificial vessels.
4. Compliance is the ability of prosthesis to elastically expand and contract in the circumferential direction in response to a pulsatile pressure.
5. Veins, which are low level pressure, return blood to the heart.
6. Arteries distribute blood from the heart to the rest of the body hence they operate with higher pressure pulses)
7. Systole – The maximum pressure reached by the fluid in the arterial system due to heart beat.
8. Diastole – The minimum pressure attained by the arterial fluid system

LIST OF SYMBOLS

D: External diameter of the diaphragm / Diameter of the piston head

d: Internal diameter of the diaphragm piston head

p_{\max} : Maximum systolic pressure

p_{\min} : Minimum diastolic pressure

S.L: Stroke length (distance traveled by the pump to displace volume of fluid from its chamber)

S.V: Stroke volume (Volume of fluid expelled from the pump chamber stroke length completed)

TABLE OF CONTENTS

DECLARATION.....	i
ACKNOWLEDGEMENTS	ii
TERMS OF REFERENCE.....	iii
EXECUTIVE SUMMARY.....	iv
GLOSSARY	viii
TABLE OF CONTENTS	ix
LIST OF ILLUSTRATION.....	xii
1 INTRODUCTION	1
1.1 SUBJECT OF THE REPORT.....	1
1.2 BACKGROUND AND SIGNIFICANCE OF THE PROJECT.....	1
1.3 OBJECTIVES.....	3
1.4 METHODS USED TO GATHER INFORMATION	3
1.5 SCOPE AND LIMITATIONS.....	3
1.6 PLAN OF DEVELOPMENT OF THE REPORT	4
2 LITERATURE REVIEW.....	5
2.1 PREVIOUSLY DEVELOPED PULSE SIMULATORS FOR HUMAN PHYSIOLOGICAL CONDITIONS.....	5
2.1.1 <i>A flow simulator developed by Holdsworth et al (1991).....</i>	<i>6</i>
2.1.2 <i>Flow simulator developed by Charara et al (1999).....</i>	<i>7</i>
2.1.3 <i>Computer-controlled pump system developed by Law et al (1987)8</i>	
2.1.4 <i>Pulsatile Flow Loop developed by Iyengar et al (2001).....</i>	<i>9</i>
2.2 CURRENTLY USED CARDIOVASCULAR SIMULATORS	10
2.3 THE CURRENTLY USED COMPLIANCE RIG	11
2.3.1 <i>Components of the Compliance rig.....</i>	<i>11</i>
2.3.2 <i>Testing methods on the Compliance rig.....</i>	<i>12</i>
2.3.3 <i>Discussion of a presently used roller pump and its control method</i>	
13	
2.4 DIFFERENT TYPES OF PUMPS THAT ARE SUITABLE FOR SIMULATION OF PHYSIOLOGICAL CONDITIONS	16

2.5	CONCLUSION BASED ON LITERATURE REVIEW.....	19
3	PROBLEM CLARIFICATION	21
4	PROCEDURE FOLLOWED TO SOLVE THE PROBLEMS	22
5	PUMP DESIGN CRITERIA	23
5.1	SPECIFICATIONS AND REQUIREMENTS OF SUITABLE PUMP DRIVER/CONTROLLER	23
5.1.1	<i>Design parameters used for the pump system for physiological flow simulation</i>	24
5.1.2	<i>Requirements for a driver/controller of a pump</i>	25
5.1.3	<i>Different pump driver / controllers evaluated.....</i>	26
5.2	OPTIMISATION OF A DIAPHRAGM PUMP BASED ON A CYLINDER POSITION CONTROLLER	27
5.3	DETAILED DESIGN OF THE DIAPHRAGM PUMP	30
6	COMPUTER BASED SOFTWARE DEVELOPMENT TO CONTROL THE DIAPHRAGM PUMP.....	34
6.1	EVALUATION OF COMPUTER-BASED HARDWARE AND COMPATIBLE SOFTWARE TO GENERATE AND CAPTURE SIGNALS	34
6.1.1	<i>Comparison of PCI30GA board and JK1 microprocessor</i>	35
6.1.2	<i>Justifications for using PCI 30GA board and MATLAB software .</i>	38
6.2	MATLAB PROGRAM DEVELOPED TO GENERATE AND CAPTURE SIGNALS WITH THE PCI30GA BOARD.....	38
6.2.1	<i>How signals are generated with the developed MATLAB graphical user interface (GUI).....</i>	39
6.2.2	<i>How signals are captured using the developed GUI.....</i>	40
6.3	APPARATUS USED TO TEST THE FUNCTIONALITY OF THE MATLAB PROGRAM AND RESULTS OBTAINED	42
6.4	PROPOSED SOLUTION AS AN INTEGRATED SYSTEM	45
6.5	DISCUSSION OF THE FINAL SOLUTIONS.....	46
6.6	EXPECTED DRAWBACKS OF THE SYSTEM	47
7	CONCLUSIONS.....	48
8	RECOMMENDATIONS.....	50

9	REFERENCE	51
	APPENDICES	53
	APPENDIX A: DIFFERENT TYPES OF PUMPS CONSIDERED	
	APPENDIX B: ANALYSES OF VARIOUS PUMP DRIVES	
	APPENDIX C: PUMP CALCULATIONS AND EXCEL SHEET	
	APPENDIX D: CYLINDER POSITIONER	
	APPENDIX E: VIVITRO SYSTEM	
	APPENDIX F: EXPLANATION OF MATLAB CODE	
	APPENDIX G: DEVELOPED MATLAB CODE	
	APPENDIX H: RESULTS FOR SIGNAL CAPTURE AND GENERATION OF DEVELOPED PROGRAM	
	APPENDIX I: QUOTATIONS AND ADDITIONAL SPECIFICATIONS	
	APPENDIX J: ASSEMBLY AND PARTS DRAWING OF PUMP	

LIST OF ILLUSTRATION

LIST OF FIGURES

Figure 2-1: Physiological pressure and flow waveforms for the human arterial tree, picture from McDonalds 4 th edition pg 179 [4].....	6
Figure 2-2: Schematic diagram of the flow simulator by Holdsworth et al (1991)	7
Figure 2-3: Block diagram of a computer controlled flow simulator developed by Law et al (1987)	9
Figure 2-4: A sketch for the pump system and the control unit for the flow loop [7].	10
Figure 2-5: A schematic drawing for the Compliance rig [9].	12
Figure 2-6: The picture of the currently used roller pump	14
Figure 2-7: The controller for the roller pump	14
Figure 5-1: Estimated velocity-time graph for an ascending aorta.	24
Figure 5-2: A figure showing the schematic of the pump chamber and its important parameters	28
Figure 5-3: Exploded view of the Diaphragm pump.....	31
Figure 5-4: Concept of a ball valve with adjustable spring stiffness	33
Figure 6-1: A graphical user interface developed for generating waveform signals and capturing of signals	39
Figure 6-2: A picture for apparatus used to test the functionality of the developed program	42
Figure 6-3: A picture showing the adapter from the PC to the signal generator and oscilloscope	43
Figure 6-4: Test results for captured signal at 1.2 kHz and generated signal at 3 Hz.....	44
Figure 6-5: Summary of solution for the design of a computer controlled pump (Not scaled).....	45

LIST OF TABLES

Table 2-1: Evaluation of different pumps that can be used for flow simulation (except currently used roller pump)	17
Table 5-1: Evaluation of different pump drivers.....	26
Table 5-2: Calculated parameters for a diaphragm pump and justifications.....	28
Table 6-1: Comparison of a PCI 30GA and JK1 Micro-controller.....	37

1 INTRODUCTION

1.1 Subject of the report

This thesis concerns the design of a linear positive displacement pump and the development of a computer control interface for such a pump. The main purpose of a computer-controlled pump is to simulate heart beats of human beings by reproducing physiologically-equivalent cardiovascular conditions such as pressure and flow-rates.

The pump will form part of an existing rig (hereafter referred to as the **Compliance rig**) that is used to test mechanical properties of natural and artificial blood vessels (grafts). The pump to be designed is to eventually replace a roller pump, which is presently used for a similar purpose mentioned above on the Compliance rig. The reason for replacing the roller pump will be discussed in the literature review.

The thesis is hence subdivided into two main sections:

- The first section is a theoretical design of a suitable pump that excludes the manufacturing process. The final deliverables of this section of the thesis is a detailed drawing of the pump as well as quotation of pump components that need to be purchased.
- The second section concerns the development of a PC controlled-interface that is to be used to reproduce physiologically-equivalent cardiovascular waveforms for pressure or flow-rate. The appropriate software has to be chosen and hence used to reproduce such waveforms. The functionality of the software is to be proven by using standard equipment e.g. oscilloscope and signal generator.

1.2 Background and significance of the Project

Testing of artificial or natural blood vessels (hereafter referred to as **grafts**) outside living organisms (*in vitro*) is important before such vessels can be implanted on individuals whose natural vessels may have been damaged or diseased. Grafts are tested *in vitro* to establish if they will be able to sustain mechanical conditions present

in human beings. Of particular interest is the compliance of grafts, a phenomenon referred to as the change of circumferential diameter of grafts in response to pulsatile pressure; as according to the AAMI standards¹. This mechanical condition is present within living organisms (*in vivo*) and it is due to pulsating arterial cyclic loading of the heart beat action [1].

Non-compliant grafts and other parameters such as frequency dependent changes in host vessels, mismatch of diameter of host vessels and grafts are reported to be the main causes of mechanical failure of grafts *in vivo* [1]. In addition, various diseases can result in a host due to mechanical property mismatch of the graft to be implanted and the host vessels. These diseases include thrombosis, intimal hyperplasia, intimal proliferation and stress increases at anastomotic sites [2]. It has been reported that graft implantation can be a means of treating certain types of intravascular diseases [3].

As a result, proper testing equipment and procedures are required to test grafts before they can be implanted into the hosts.

One of the requirements of the Compliance rig, which is used as the test equipment, is to be able to accurately reproduce pulsating flow of fluid (similar to physiological conditions) in a closed fluid circuit (The human physiological pressure and flow waveforms shall be presented in the Literature review). This requirement is to be met by a pump that forms part of the rig. The ease with which the physiological conditions are to be reproduced is also an important characteristic of the Compliance rig, and the presently installed **roller pump** doesn't meet this requirement. Based on the above information there was a need to the design and development of a computer-controlled pump as requested by the CVRU.

¹ AAMI is an American National Standard for Cardiovascular Implants-Vascular prosthesis

1.3 Objectives

The project is meant to form the basis on which future work can be done in realizing a computer-controlled pump to test grafts. As a result, objects that are to be met on this thesis are to:

- a. Determine the optimum design of a pump to allow for proper testing of the grafts under physiologically-equivalent conditions. This pump is to eventually replace the existing roller pump.(A detailed drawing of the optimum design of a pump is also required to future manufacturing)
- b. To develop software program that can be used to easily input physiological waveforms to drive pump system. The waveforms must also be reproducible.

1.4 Methods used to gather information

Various methods were used to gather information necessary to execute the project:

- a. An internet search was performed to source for commercially available pumps used for simulation cardiovascular conditions.
- b. Consultation with research companies
- c. Consultation with supervisors
- d. E-mail correspondence and telephone discussion with software engineers at EAGLE TECHNOLOGY regarding programming of a PCI multi-channel card.
- e. E-mail correspondence with the MATHWORKS support group concerning MATLAB programming.
- f. Telephone conversations with engineers from FESTO company

1.5 Scope and Limitations

The main limitation of the project was that there was no budget allocated hence readily available devices had to be used in order to perform practical tests of the developed MATLAB program.

1.6 Plan of development of the report

The report begins by undertaking literature review where different pump systems used to reproduce physiological parameters in order to test mechanical behaviour of natural and artificial vessels (grafts) are studied.

Problems to be tackled in the thesis are then more clearly explained in section 3 (Problem clarification) General procedure used to gather information is then presented.

Pump design criteria is then undertaken where requirements and specifications for suitable pumps are stated. Suitable pump drivers are presented and detailed design of diaphragm pump is presented

A computer based software development is then looked at with various options determined. Results conducted on MATLAB program conducted are presented. Finally all proposed solutions of the project e.g. pump driver, controller and software and pump type are shown how they could be integrated into one unit. Based on the findings of the thesis, conclusions are drawn and recommendations are made.

2 LITERATURE REVIEW

In this section of the report, background information of the work done relating to the development of pumps that have been used simulate cardiovascular conditions is given. The current compliance testing rig, on which the pump to be developed is to be incorporated, is also discussed. Results of research on currently used pulsatile pumps are also presented.

Finally conclusions are drawn based on the literature review to highlight challenges faced by designers to reproduce cardiovascular / physiological conditions. The conclusion is intended to clarify how the solutions developed on this thesis will form the basis on which a fully computer-controlled pump for physiologic flow simulation can be realized for the CVRU in future.

2.1 Previously developed pulse simulators for human physiological conditions

As previously explained, diseased or damaged blood vessels in living organisms are normally replaced with suitable grafts. Hence there is a need for testing the grafts with appropriate equipment before implantation. An ideal flow simulator is desired to be able to reproduce physiological pressure and waveforms similar to those represented in figure 2-1 below. The basis on which these waveforms can be programmed on a PC is that since they are continuous and periodic functions, they can be represented by a Fourier function given by the following equation [4]:

$$F(t) = \frac{1}{2} A_0 + A_1 \cos \omega t + A_2 \cos 2\omega t + A_3 \cos 3\omega t + \dots + A_n \cos n\omega t + \dots \\ B_1 \sin \omega t + B_2 \sin 2\omega t + B_3 \sin 3\omega t + \dots + B_n \sin n\omega t \dots \dots \dots [4]$$

Where:

- Angular frequency, $\omega = 2\pi T$ and T is the period of the function, F (t).
- Constants A_0 to A_n and B_0 to B_n represent Fourier constants that are selected to represent any required waveform.

The main advantage of being able to reproduce similar waveforms below is to be able to test grafts under similar conditions that they will be subjected to *in vivo*.

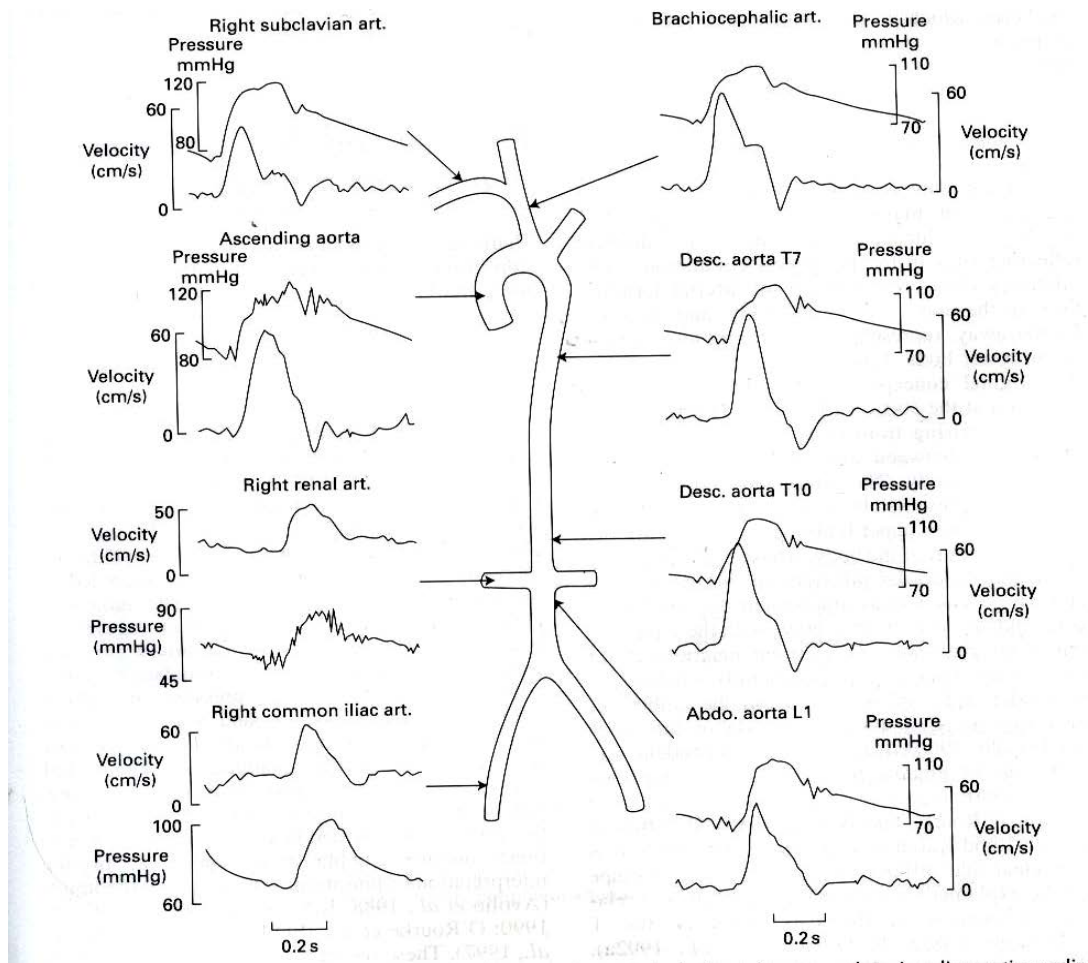


Figure 2-1: Physiological pressure and flow waveforms for the human arterial tree, picture from McDonalds 4th edition pg 179 [4]

The following sections of the report shall briefly explain some of the systems developed to reproduce physiological conditions in order to test grafts. The explanation of various pulse simulators will place more emphasis on the types of pumps used on the system and the control methods used for reproducing physiological flow conditions. Although some of the pulse simulators were used to test heart valve operations, the key feature of such simulators is that they are also capable of producing pulsatile flow.

2.1.1 A flow simulator developed by Holdsworth et al (1991)

(As according to reference [5])

The design is reported to be capable of producing various pulsatile flow waveforms. The pump system consists of a two piston pumps driven by a single rack-and-pinion

stepping motor as shown in figure 2-2 [5]. The motor controller controls each micro-step of the rack such that one motor micro-step results in a volume displacement of the cylinder by **0.51 μ liter**. The net effect is that flow-rate of the pump is controlled. In order to generate physiological flow waveform, the designers programmed flow-rate as a function of time. This was because the relationship between the angular displacement of the motor shaft and volume displaced from the pump was known (i.e. 0.51 μ l per micro-step of rack of the motor). The programmed function was digitized and loaded to the motor controller to drive the motor.

The other key feature of the flow simulator is that it consists of a controllable valve, which ensures one piston refills while the one displace fluid. The other importance of the valve is that it can allow reverse flow unlike passive valves that allow fluid flow in one direction only. Reverse flow does occur in human blood flow hence the use of controllable valve mimics human physiological blood flow more closely.

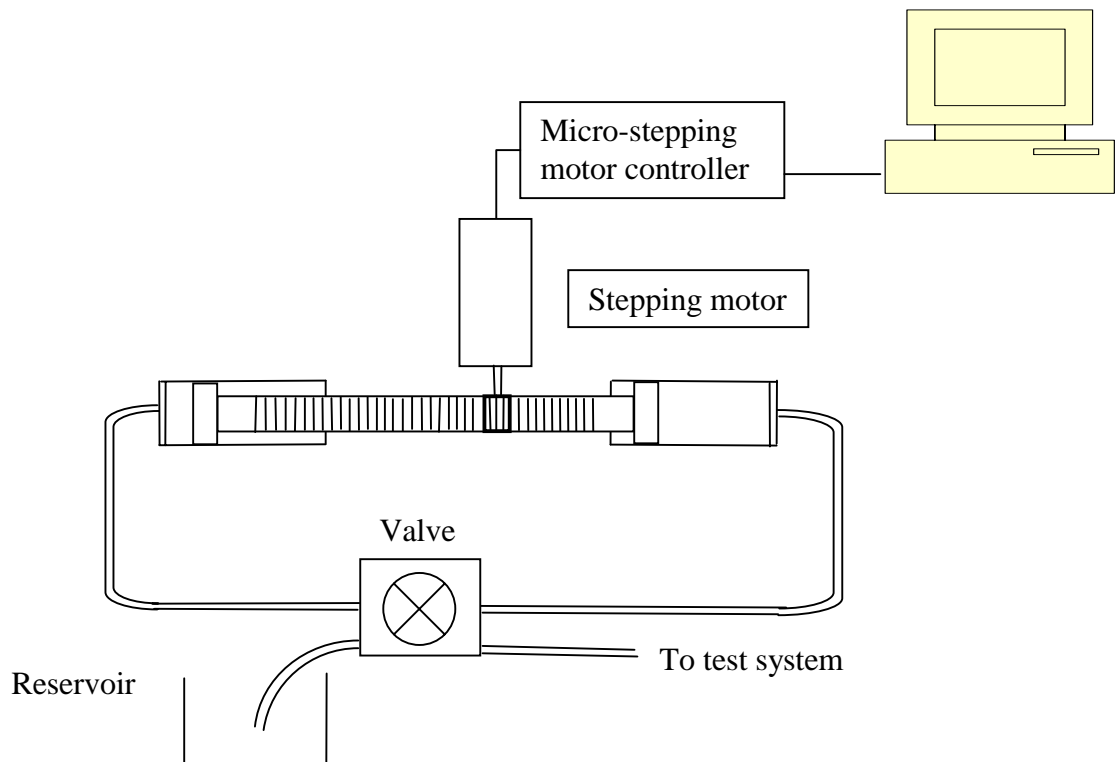


Figure 2-2: Schematic diagram of the flow simulator by Holdsworth et al (1991)

2.1.2 Flow simulator developed by Charara et al (1999)

(The explanation below is according to reference [2])

The pump system used, which is a rack-and-pinion driven piston pump, was the same one developed by Holdsworth *et al*, which is shown in figure 2-2 above.

In order to reproduce physiological waveforms, a transfer function² for the complete flow simulator system was determined for different amplitudes and frequency.

To ensure that the system generated desired waveforms, the input signal to the system was compared to the measured flow signals. The correction of the input signal was hence made based on its deviation from the measured signals. The corrected signal, **which is reported to be a flow-wave pattern**, was digitized and used as an input to the power amplifier, which in turn drove the pump. Again as with the simulator developed by Holdsworth *et al*, this suggests that flow rate is controlled not pressure. Physiological flow waveforms were reproduced successfully according to designers. However the duplication of physiological pressure waveform weren't exactly achieved. The reason for the non-duplication of pressure waveforms, according to the designers, is due to the fact that pressure does not only depend on the imposed flow rate but also on the elasticity of both the tubing and the grafts used for the flow simulation circuit. Because the elasticity of the tubing and graft used in flow simulation circuits is different from that of natural vessels within living organisms, the pressure waveforms will therefore be different.

2.1.3 Computer-controlled pump system developed by Law *et al* (1987)

(According to reference [6])

The system utilised a roller pump that was driven and speed-controlled by a stepper motor and its controller respectively. A block diagram below illustrates the entire system. The stepper motor produced 400 steps per revolution.

² A transfer function is a Control Theory function that relates the input signal of the system to the output response of the same system.

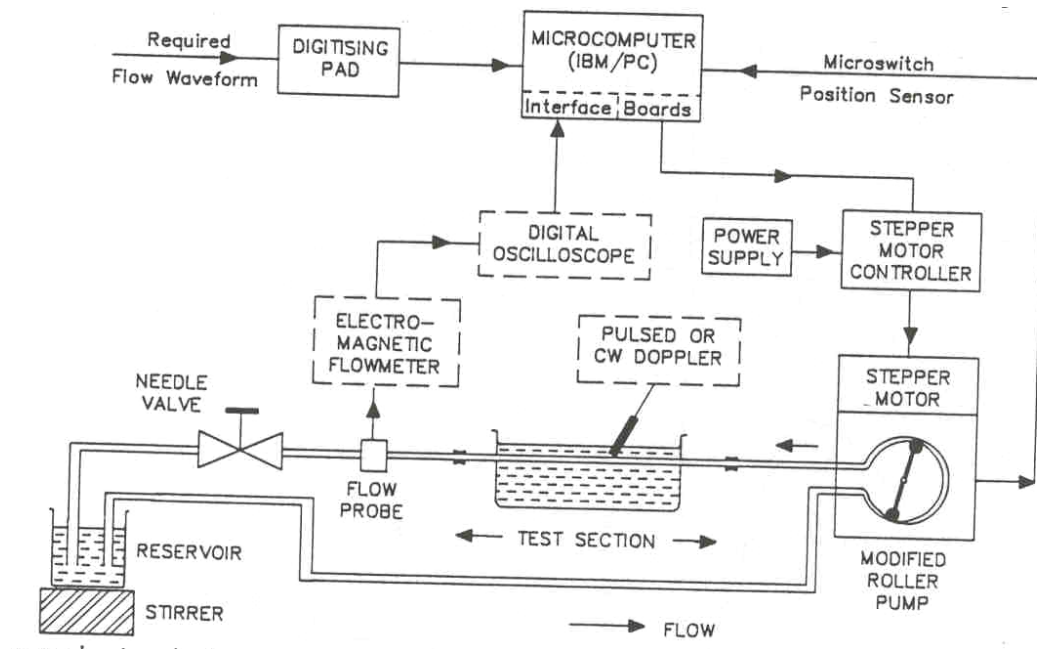


Figure 2-3: Block diagram of a computer controlled flow simulator developed by Law et al (1987)

In an attempt to generate physiological flow rate, metal inserts were used between the tube and the rollers to adjust the degree to which the tube is compressed.

Flow-rate was then measured with a probe and flow signals were fed to a stepper motor controller to adjust the pump speed through the feedback system shown on the diagram.

The other key feature of the system includes the use of an adjustable needle valve to adjust the flow resistance so that it (flow resistance) doesn't influence the waveform being generated.

2.1.4 Pulsatile Flow Loop developed by Iyengar *et al* (2001)

(This is as according to reference 7)

This pulsatile flow loop was mainly developed for testing heart valves. It is reported to be capable of producing physiological aortic pressure waveforms. The pump system consists of a piston-cylinder system, which was used to create required pressure conditions within a fluid filled chamber. An electro-magnetic voice coil was used control the motion of the piston via a computer. A sketch for the pump system with the chamber is shown below:

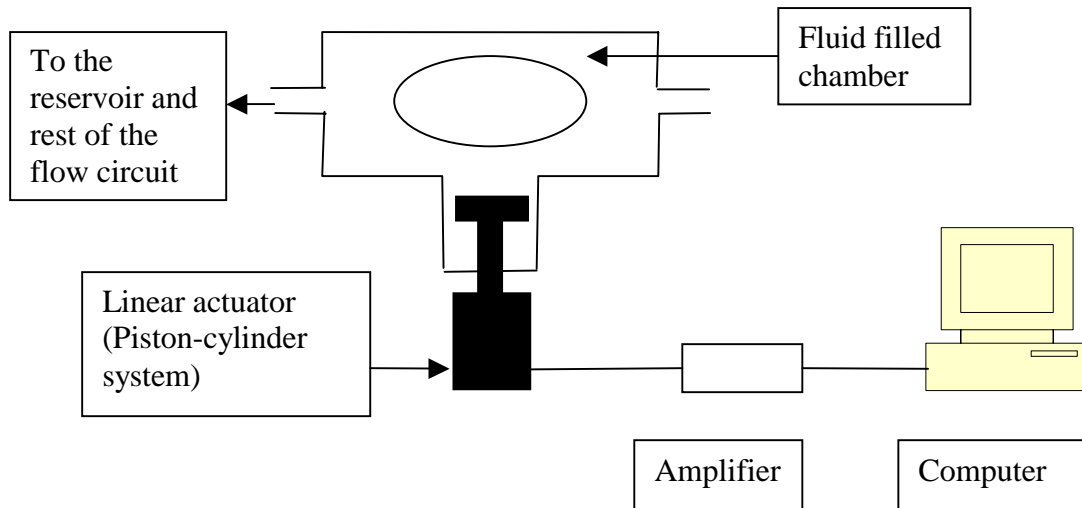


Figure 2-4: A sketch for the pump system and the control unit for the flow loop [7].

The pressure on the chamber is adjustable by a height of the liquid in the reservoir that forms part of the flow loop circuit [7]. It is further stated that a check valve is placed between the reservoir and the fluid filled chamber to prevent any reverse flow to the chamber when the pressure is being raised in the chamber.

2.2 Currently used cardiovascular simulators

Literature survey done, revealed that very few pump systems specifically intended for cardiovascular simulation are not easily available on the market. An e-mail correspondence with *Marcel C.M. Rutten from Eindhoven University of Technology* revealed that indeed very few companies currently sell such pump systems. Rutten et al manufactured a bioreactor/pulse simulator themselves. Their system generates cardiovascular conditions but it is mainly used for dynamic aortic culturing of valves. Companies that were revealed to have been selling required pump systems were *Numatics* and *Vitro Systems*. From research done only the latter company was found to be still existent. They had a system that was reported by the sales engineer that it is capable of producing required cardiovascular conditions. See **APPENDIX E** for details.

2.3 The currently used Compliance rig

This section shall briefly explain some of the components of the Compliance rig and the testing methods that are conducted on the rig.

2.3.1 Components of the Compliance rig

The figure below (figure 2-5) shows the components used in the current Compliance rig. Some of the components of the Compliance rig are explained below:

Water bath (WB)

The water bath houses the graft sample that is being tested. The water in which the graft is submerged is normally kept a temperature of 37⁰C to mimic human conditions. A heater element is used for this purpose.

Instrumentations and data acquisition

Presently only pressure sensors are installed on the distal and proximal side on the graft sample to measure pressure. Digital Camera and microscope, used for diameter measurement of the graft are not currently installed. Flow probes, for measuring fluid flow are also not installed at the moment. A triton amplifier (TA) is used to amplifier signals from the sensors (only the pressure transducers are installed at the moment) before they can be send to the *PC30 MIA digital-to-analog board* (Eagle Technology, South Africa). The card is used to acquire such signals and display them on the PC monitor with the *Waveview 1.4* software (Eagle Technology, South Africa).

Windkessel chambers

These chambers, shown by W1 and W2 on the Compliance rig sketch, are used to mimic the elasticity of the aorta within the human body. They are therefore used to smoothen an otherwise sharp increase in systolic pressure produced by a roller pump [8].

Needle valves

They are used to set distal and proximal pressures depending on where they are located relative to the graft sample.

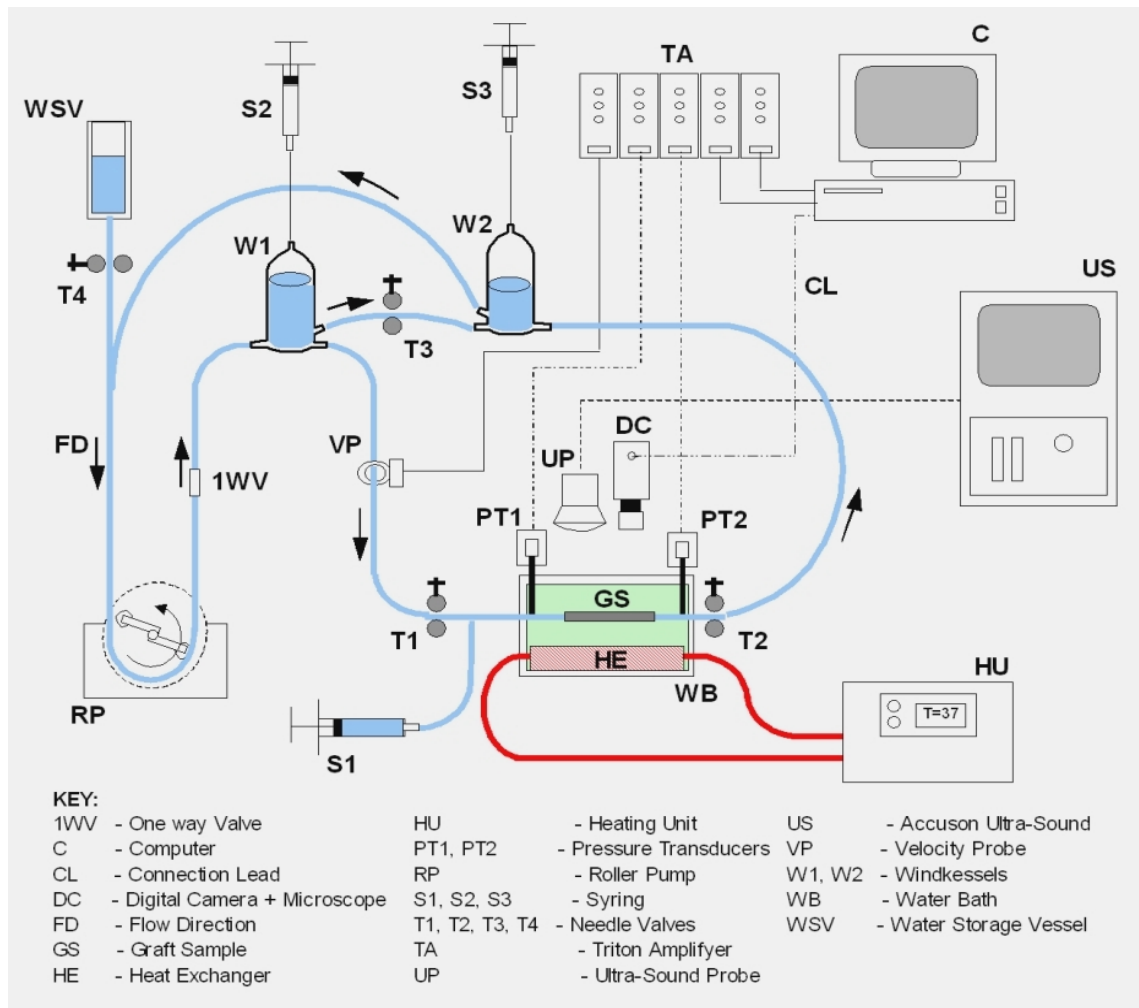


Figure 2-5: A schematic drawing for the Compliance rig [9].

2.3.2 Testing methods on the Compliance rig

The compliance rig is used to test grafts statically and dynamically. Both methods are used to establish the compliance of grafts. In carrying out the static tests, an incremental pressure is induced on the inside layer of the graft. A syringe, which is shown on figure 2-5 as “S1”, is used to apply the required internal pressure. The static test is used to determine how the external diameter of the graft changes in response to the applied internal pressure. During static testing, the pump that is used for producing pulses of fluid is stopped.

To perform dynamic tests, pulses of fluid are induced on the grafts by a peristaltic or roller pump over a certain period of time. (The pump is shown as “RP” on figure 2-5). The objective is to simulate more realistically the cardiovascular conditions found

within a human being. The difference between static and dynamic compliance is related to the strain rates [8]. While static testing doesn't account for strain rates of grafts, dynamic testing does since pulse rates (produced by the pump) are directly proportional to the strain rate [8].

2.3.3 Discussion of a presently used roller pump and its control method

This section of the report briefly explains the function of the roller pump and hence provides reasons for its intended replacement.

Roller pumps, which are also known as peristaltic pumps, are positive displacement types of pumps that consist of a length of tubing inside a curved raceway. A roller pump consists of two rollers that are eccentrically rotated by a motor. The rollers continuously squeeze the fluid inside a flexible tubing to provide pulsatile flow. See picture of the pump below in figure 2-6.

In an attempt to produce physiological flow waveforms, inserts are placed between the raceway between the raceway and the flexible tubing as shown in figure2-6 below. Physiologically the purpose of the insert is two-fold. The front portion of the insert is to create a sharp increase in pressure, which is similar to a systolic pulse. The tapered end of the insert is to allow for gradual decrease in pressure to a diastolic value of about 40mmHg assuming normal systolic pressure of 120mmHg [8].

A roller pump without inserts was also found to produce a significantly reduced flow-rate since the uncompressed tube will quickly expand after the roller has lost contact with the tubing. This quick expansion contributes to a decrease in flow [6]. Without inserts it was also found to produce pure sinusoidal waveforms, not physiological waveforms represented by figure 2-1 in the previous section of the report. [8]

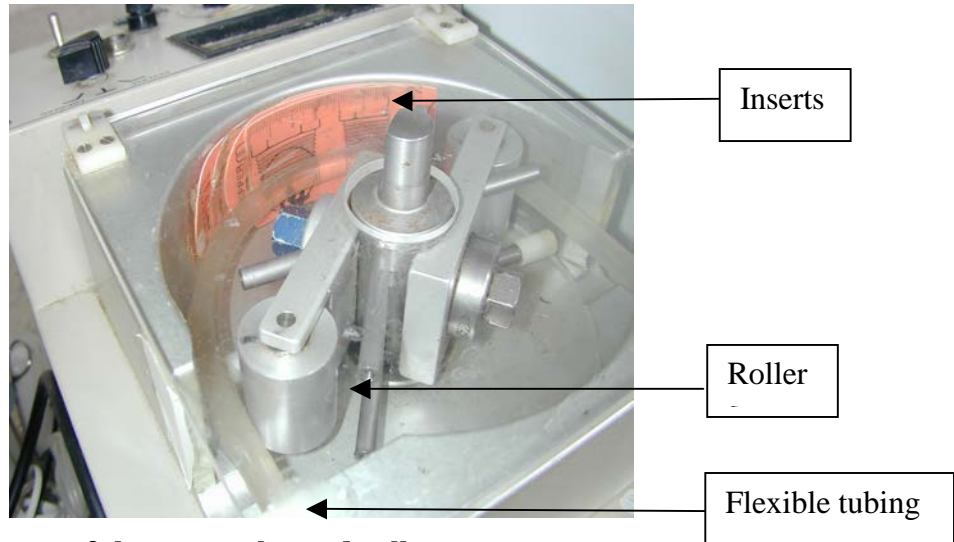


Figure 2-6: The picture of the currently used roller pump

Roller pump controller

The controller shown in figure 2-7 below is used to set the motor speed at which the roller pump is rotated. The speed of the motor is specified by beats per minute (bpm) and the normal speed used is 72bpm since this represents the average heart beat rate.



Figure 2-7: The controller for the roller pump

Advantages and disadvantages of the roller pump

The advantages of the pump include the following:

- It can be easily sterilized due to the circular shape of the tubes [18].
- For a particular insert, which can be difficult to position, acceptable pressure waveforms can be achieved.
- They require no valves and hence simplifies the design
- Metal parts of the pump do not come into contact with the fluid e.g. rollers. This allows for only the material of the tubing to be selected according to the design requirement e.g. corrosion resistance/sterilization/chemical inertness.

The main disadvantage of roller pumps is the difficulty in reproducing different waveforms (programmability) due to the adjustments of inserts (Trial and error approach). The other disadvantages are as follow:

- As the tubing wears out, flexibility is lost and inaccuracy results.
- The elasticity of the tubing reduces the pump's impedance and hence allows waveforms to be affected by load [18].

Based on the analysis of the roller pump, the main disadvantage that motivated the proposal of the project was the use of inserts to reproduce pressure pulses. The method used was found to be very inflexible and required trial and error methods to generate physiological waveforms. An easier and reproducible method was required to generate physiological waveforms hence the motivation of this project to develop a computer-controlled pump.

2.4 Different types of pumps that are suitable for simulation of physiological conditions

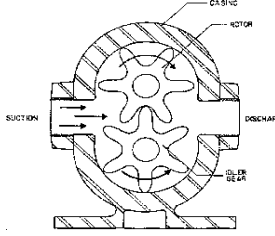
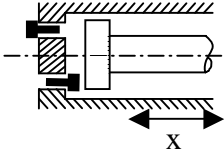
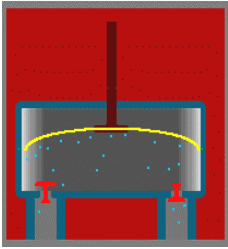

Pumps that have been popularly used to simulate pulsatile flow are of positive displacement type. Examples of such pumps include: gear pumps, roller pumps, diaphragm pumps and mechanical piston pumps.

However other types of pumps including centrifugal pumps, ventricle pumps have also been used for physiological flow simulation. For full definition and explanation of the above-mentioned pumps, see **APPENDIX A**.

The main advantage for using positive displacement pumps is that they are easily controllable. This is due to the fact that their output i.e. displaced volume can be easily related to input displacement (e.g. in case of piston pumps). This was shown in the system designed by Holdsworth *et al* [5], which used a piston pump. Based on the above statements, different types of positive displacement pumps were investigated. These were diaphragm pumps, mechanical piston pumps, gear pumps and roller pumps.

A summary of different types of pumps mentioned above is given in tabulated form below. Specific methods used to compare the different types were: Mechanical operation of the pump, Methods of driving the pump, relationship between displaced volume and input displacement. Main advantages and disadvantages of each pump were also listed. The relationship between volume and input displacement was found from described simulators that it determines how easily the pump can be controlled.

Table 2-1: Evaluation of different pumps that can be used for flow simulation (except currently used roller pump)

<p>Pump type</p> <p>Method of evaluation</p>	<p>Gear pump</p> 	<p>Mechanical piston pump</p> 	<p>Diaphragm pump</p> 	<p>Ventricle pumps</p> 
<p>Operation</p>	<ul style="list-style-type: none"> -Rotation of gears -Fluid is displaced from inlet to outlet via meshing of gears. 	<p>The bi-directional motion of piston (x) causes inlet and outlet of water via valves</p>	<ul style="list-style-type: none"> - Piston attached to diaphragm (elastic membrane) - Fluid intake and outlet takes place via passive valves 	<p>-A chamber divided into 2 parts: One for fluid chamber and valves and another section is for air.</p>
<p>Possible drivers</p>	<p>A rotating motor</p>	<ul style="list-style-type: none"> - Linear motors - Ball screws -Piston-and-cog driven by rotational motor - Pneumatic drives -Rack-and-pinion systems -Cam driven piston with rotational motor -Electro-magnetic voice coils <p>N.B: All motors that can be used must be stepper motors or servo motors since these are easily programmable</p>	<p>Similar to mechanical piston types since they're both actuated from a piston rod</p>	<p>Compressed air can be used to provide air pulses</p>

Relationship of displaced volume and driver input	No definite relationship can be made between input rotation and displaced volume due to gears	Linear relationship between steps input (stroke) and displaced volume.	Due to the elastic material a non-linear relationship between step input and displaced volume is not achieved but there is a proportional relationship	The system is difficult to control since it is difficult to relate input air pulse to displaced volume
Main advantages	Requires no valve design	Flow-rate can be easily controlled due to linear relationship between stroke and displaced volume. Actuators that readily provide linear motion make the control of the pump much simpler e.g. linear motors	-No leakage problems since a piston head is replaced by an elastic membrane. -No sealing required	
Main Disadvantages	-Difficult to lubricate gears since fluid used testing grafts is non-lubricating. -Damage of haemodynamic characters of fluid due to gear meshing [2], [5], [6].	-Leakage losses between cylinder and piston can occur - Even sealing between piston and cylinder suffers from wear in a long run.	-Requires design of valves -Elastic membrane causes non-linear relationship between output fluid and strokes (This results in loss of energy)	Lacks flexibility to program different waveforms [8]

(Diaphragm pump picture from: www.maintenanceresources.com/.../Pumps/diaphram.htm)

(Gear pump picture courtesy of KU magazine, www.goae.go.th)

From the analysis of different types of pumps, the piston and diaphragm pumps were to be further investigated as they can be easily controlled. Since the currently installed roller pump is not easily controllable, the identified pumps can provide an optimum solution for the replacement of a roller pump.

2.5 Conclusion based on Literature Review

It is better to develop own design of pulsatile flow system than to purchase the system

Literature survey undertaken indicated that very few cardiovascular simulators are commercially available on the market and researchers developed most systems themselves. [For systems that were readily available on the market, these were found to be very expensive (~\$18 000, US Dollar, *Vitro system*). This typical system can be seen in **Appendix E**. The pump system and controller is believed to meet cardiovascular requirements. The drawbacks with this system is that

1. It still requires appropriate waveforms to be programmed
2. It has a high selling price

The above drawbacks provided further justification self-develop the required pump system instead of purchasing one. The system to be designed is hence expected to cost less than the above system.

Requirements for ideal flow simulators are challenging

Also based on literature review conducted on previously developed computer based flow simulators, it is clear that the development and realization of a computer-controlled pump for simulation of physiological conditions requires the fulfillment of a number of technically challenging requirements. The fulfillment of such requirements presents technically challenging tasks and in order to successfully accomplish those tasks more research time must be allocated and is certainly more than undergraduate thesis time.

Some of the key requirements of physiological flow simulators that were identified are listed below [2, 5 and 6]:

- (a) They must be capable of producing a wide range of flow-rates
- (b) They are to be easily programmable to produce a variety of pulsatile flow rates including reverse flow. (This was not found to be the case with many systems as they were only capable of producing limited

types of flow waveform. Pressure waveforms were not completely duplicated)

- (c) Continuous pulsatile flow without pressure/energy losses must be maintained by the system.
- (d) The system must behave like an ideal source and should produce sufficient pressure. The pressure waveforms are not to be affected by flow resistance due to tubing of the circuit as possible.
- (e) Valve design of pumps is also separate challenge as their operation can affect required waveforms.
- (f) An optimum design of a pump to closely mimic physiological conditions.

Piston driven positive displacement pumps are ideal for flow simulation

Based on the analysis of different types of pumps that can be used for physiological flow simulation, piston and diaphragm pumps were found to meet most of the above-mentioned requirements. The linear motion of piston provides the strokes of these pumps. As a result, these two concepts were to be evaluated further in the main section of the report.

As mentioned that the accomplishment of all the requirements for a flow simulator will require more time, only two main tasks were chosen as focus of the thesis. The two tasks are:

- i. The selection and optimization of a suitable pump that can be used on the Compliance rig to eventually replace a roller pump.
- ii. To develop a program that will form an input to the pump system so that eventually any physiological waveform can be generated on the Compliance rig.

Therefore the accomplishment of the above two tasks, especially the program, will form basic starting points from which a computer controlled system can be accomplished in future.

3 PROBLEM CLARIFICATION

This section of the report is meant to clarify intended achievements in this project. It is also intended to further clarify how the tasks mentioned in the literature review shall be accomplished.

The realization of a complete computer-controlled pump for cardiovascular flow simulation requires various tasks to be tackled as outlined in the literature review. However due to the limited time frame of the thesis and the technical know-how, only the following problems shall be tackled as the solutions to such problems could set up a starting point from which a fully developed computer-controlled pump can be realized through further engagement in the project.

The clarified problems that were to be specifically tackled on this thesis were to:

- a. Select, optimise and produce detailed drawings of a pump that can eventually replace the existing roller pump. A suitable pump controller is also to be identified.
- b. Identify a computer based hardware and a suitable software program and hence use the combination of the two to develop a program that can generate any waveform signal and capture signals to display them to the user.

Generated signals are intended to form an input to the pump controller while the ability of the program to capture signals will be important to capture pressure/flow signals on the Compliance rig.

4 PROCEDURE FOLLOWED TO SOLVE THE PROBLEMS

The key characteristic of the two identified problems in Section 3 is that they are all dependent on each other. This is due to the fact that solutions to such problems (i.e. devices to be identified) will be integrated into the existing Compliance rig to achieve a common task, which is to simulate cardiovascular flow conditions. The integration of devices into an existing rig is however beyond the scope of this project.

The method used to arrive at the solutions for the stated problems began with the analysis of various classes of pumps that can be used on the Compliance rig. Advantages and disadvantages of each pump were listed and based on the requirements of the pump an appropriate pump was chosen.

Before the pump was optimized to determine its critical dimensions, different methods of controlling the pump were considered. Based on requirements listed for a pump controller, a suitable controller for the pump was identified. The parameters of the pump were chosen based on the specifications for the chosen controller.

A software and relevant computer hardware that could interface to the chosen controller were identified. Since all the above-mentioned tasks were theoretical, the practical requirement of the thesis was to develop a program with the chosen software.

The development of the program involved the study of the software manuals. Once the program was developed, it was tested with standard equipment in the laboratory.

5 PUMP DESIGN CRITERIA

This section of the report will describe the design criteria used to select the appropriate pump and suitable pump driver (pump system). This begins with the identification of specific parameters that have to be met by the pump system. The specific parameters are related to physiological parameters e.g. pressure, frequency and flow-rate.

As discussed in the Literature review, piston and diaphragm pump were chosen as viable options for replacing the roller pump on the existing Compliance rig. However justifications for further development of the diaphragm pump are given.

Three viable pump drivers from **Table 2-1** (Literature review) are analysed further and the best pump driver for the diaphragm pump is chosen.

Based on the specification of the chosen pump driver and specific physiological parameters that have to be met by the pump system, **critical parameters** for the diaphragm pump were determined. (Equations and Excel calculations are presented in **APPENDIX C**) An excel spreadsheet also include calculations for the piston as it was used as the starting point for optimizing the diaphragm. The reason for this is that piston pumps are simpler in geometry compared to diaphragm pumps. (Diaphragm pumps are also special types of piston pumps). This was also to identify ballpark regions of anticipated power, force and speed of the pump.

5.1 Specifications and requirements of suitable pump driver/controller

In this section, specifications for a pump and its controller/driver are highlighted. Various ways of driving / controlling the pump are also presented and the suitable type is chosen.

Most requirements for a pump and its driver / controller are related to the physiological parameters found in human beings since the pump is to be used for a similar purpose.

5.1.1 Design parameters used for the pump system for physiological flow simulation

Before choosing a suitable pump system, specific parameters that the pump system is to be designed for are mentioned. They are presented below:

Pump speed and flow-rate

The speed of the driving mechanism is important because it will determine how quick the pump will complete required stroke length. The speed of the pump is related to the heartbeat rate of human being and is measured in beats per minutes (bpm). The average heart beat rate is 72 bpm and is equivalent to 0.83 seconds. The pump will however be designed for **180bpm** (about 0.33 seconds), which is a maximum value reported heart beat rate [2].

The pump system critical time at which the entire fluid should have been displaced from the pump chamber is to be **0.1seconds**. This is the time at which maximum velocity is reached as measured in the waveform of the *ascending aorta*. It is indicated by the graph below:

Graph for fluid speed versus time of the ascending aorta in a human being [4]

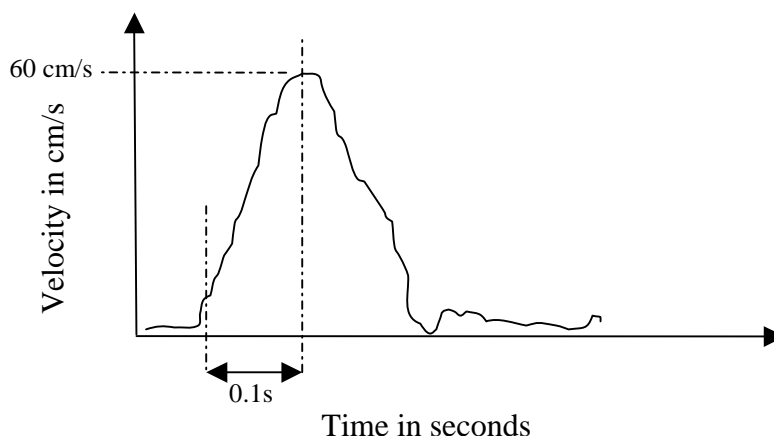


Figure 5-1: Estimated velocity-time graph for an ascending aorta.

The normal stroke volume for an average heart is estimated to be 80ml [10]. The pump will be designed for maximum stroke volume of **100ml**. This value is chosen to accommodate any losses that can occur on the Compliance rig circuit. Since it is

desired that the system should displace the total volume of 100ml in 0.1s, the flow-rate for the pump is to be **1000ml/s (1 liter-per-second)**

Pressure

The pump is to produce a maximum pressure of **250mmHg**. An average pressure is 120mmHg but for patients with Hypertension, this pressure can rise significantly. A pressure of 250mmHg is hence used.

5.1.2 Requirements for a driver/controller of a pump

A suitable driver for the pump must fulfil the specifications for the pump stated above. In particular, the most important requirement of a suitable driver is to be able to respond quickly to its input. The different types of inputs can be an electric signal, pressure signal.

The controller must ensure that the pump can achieve frequency of 3Hz (0.33s). The quick response time of the pump driver is also necessary to accurately reproduce physiological waveforms. A comprehensive list of requirements for a pump driver / controller is listed below:

- i. A linear relationship between the flow-rate of the pump to the step input of the driving mechanism is desirable. (This will result in minimal energy being lost from the input energy supplied. The supply energy could be electrical energy, hydraulic energy, etc.)
- ii. A driver that readily produces linear motion to the pump is also desirable for reasons similar to the one mentioned in (i).
- iii. Fast response of controller to its input is necessary in order to accurately reproduce the pressure and flow waveforms.
- iv. It must be able to produce a flow rate of **1 liter per second**. (As specified in **section 5.1.1**)
- v. A readily available and an economic option must be provided without compromising the requirements mentioned in (i) to (iv).

5.1.3 Different pump driver / controllers evaluated

Table 5-1: Evaluation of different pump drivers

Solution		<i>Cam-driven piston</i>	<i>Rack-and-pinion</i>	<i>Pneumatic cylinder positioner</i>
Requirements				
1	Displaced volume to input signal relation	Each flow waveform requires specific cam design. Computer aided design is needed	The step input of the stepper motor can be easily related to displaced volume	A relationship between the input voltage signal and the cylinder movement can be established hence pump displaced volume
2	Response time	Electrical signals response are related to motor type	Stepper motor response is also related to electrical response	Air has quick response time
3	Energy losses while in operation	Losses of input energy due to cam & follower frictional contact	Meshing of gears between rack-and-pinion result in noise and frictional losses	Losses can result from leakage of air
4	Ability to interface to electrical controller	Various motors e.g. dc motors, induction motors have electrical controllers	A stepper motor and controller (Stepper motor are easily programmed	Due to the e/p converter, the positioner can be controlled by programmed voltage signals
5	Availability	The manufacturing of cam by Computer Aided Manufacture can be done by Mechanical Engineering workshop	Rack-and-pinion systems are commercially available.	South African companies like RS Components, FESTO supply these products.

Final choice for the pump driver and justification

From the different drivers that have been analysed in the above table, the Pneumatic cylinder positioner was selected as the best concept based on the quick response. The integration of the unit into the Compliance rig can also be done with ease because compressed air is readily available in the laboratory. In **Appendix B**, a detailed explanation of different pump drivers is given. Since the Cylinder positioner consists of an electrical interface, suitable software (to be selected in **Section 6** of the report) can be used to program waveforms signals that can form an input to the cylinder positioner.

5.2 Optimisation of a diaphragm pump based on a cylinder position controller

The main purpose of this section of the report is to highlight the important parameters of a diaphragm pump that were calculated based on the specifications in **section 5.2.1** and cylinder positioner specifications. (See Appendix D).

Details of formulae and method used to optimize a diaphragm pump are given in Appendix C. An excel spreadsheet, also in Appendix C, was used to perform calculations. This section will therefore only highlight the final choices of parameters based on those calculations.

Following from the schematic drawing in (figure 5-2) for a diaphragm pump, parameters that were determined to be critical are:

- Small diameter, d , for piston attachment.
- Diameter (D) for the elastic membrane
- The stroke length, (S.L): This represents the maximum displacement of the piston required to displace the specified stroke volume, SV , of **100ml**(See **section 5.2.1**)
- Vertical angle, θ_{max} , of the elastic membrane when the piston has fully retracted in order to fill the pump chamber.

The above-mentioned parameters were then used to calculate the maximum piston speed, power and the expected tube velocity on the Compliance rig.

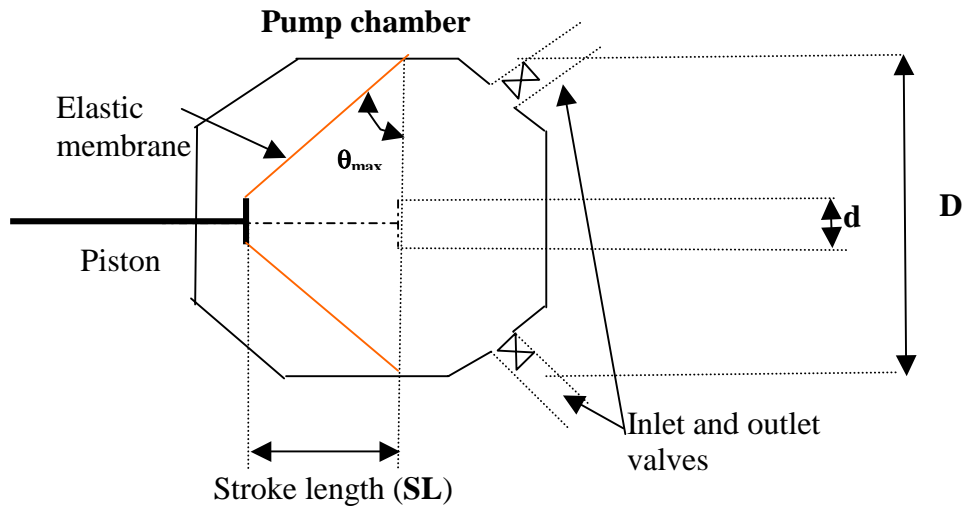


Figure 5-2: A figure showing the schematic of the pump chamber and its important parameters

Chosen parameters of the pump based on Excel calculations in Appendix C.

Table 5-2: Calculated parameters for a diaphragm pump and justifications

Parameter	Calculated/optimized value	Justification
Piston diameter (d)	25 mm	
Elastic membrane diameter (D)	110mm	As the piston velocity was desired to be very high hence the stroke length must be as short as possible. By minimizing stroke length to 25 mm (a minimum specified stroke length for a cylinder positioner, see Appendix D) parameter decreased as a piston velocity increased.
Maximum angle, θ_{max}	30°	The calculation for the angle was based on “d” and “D”.
Stroke length, SL	25 mm	Based on calculations, the value for the stroke length satisfied requirement for the stroke volume of 100ml .
Speed	0.24 m/s	
Power	0.033kW	This power is sufficient to generate the required maximum pressure of 250mmHg . The positioner

		can produce maximum power of 2.5kW hence it will meet the required maximum power.
Volume flow rate	1 litre/s	The cylinder positioner is capable of producing flow-rate of 4.2litre/s (maximum)
Velocity at the tube	2m/s	This velocity was based on the tubing diameter presently used. However the speed can change for different tube diameter.

5.3 Detailed design of the Diaphragm pump

The detailed design of the diaphragm pump will look at the material requirements of the pump, valve, manufacturing and assembly of the pump. Detailed drawings of the pump are presented in Appendix J.

Material consideration

The main requirements for suitable materials to be used for the design of the pumps are that:

- i. They must ensure chemical stability during sterilization [2].
- ii. They must also have corrosion-resistant properties [2].

Suitable materials that meet the above requirements are:

- Stainless steel
- Teflon
- Plexiglas
- Perspex
- Polyvinyl chloride
- Silicon

Stainless steel, Teflon and silicon (medical grade) are suitable to be used for various components of the diaphragm pump.

Valves

The valve that was to be used for this application was to be passive i.e. to close and open automatically in response to the pressure difference between the inlet and the outlet.

Different components of a final design of a diaphragm pump

In this section, different parts of the designed diaphragm pump are briefly explained. Emphasis is placed on dimensions and material that can be used to manufacture the parts.

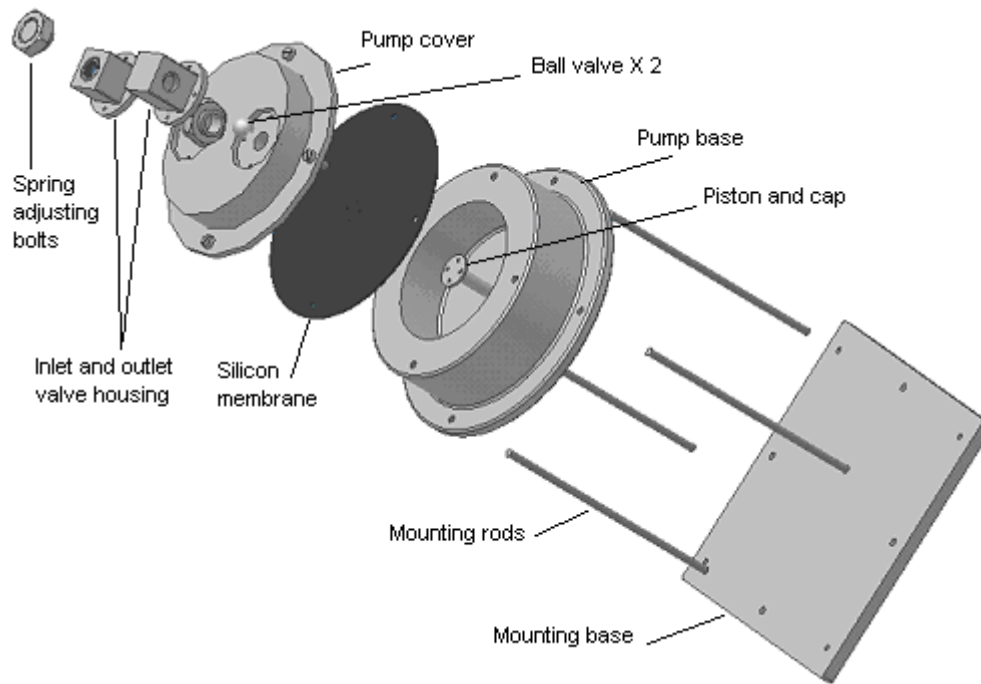


Figure 5-3: Exploded view of the Diaphragm pump

Pump base

The pump base houses the silicon membrane and the piston rods. Critical dimensions of the feature are its internal diameter and the inside depth. The internal diameter was optimized to be 110mm while the inner depth was to be more than the optimized stroke of the pump. Stroke of the pump was optimized to be 25 mm hence the inner diameter was set to 40mm.

The material chosen for pump base was stainless steel. Specific grade can be decided during manufacturing stage.

Elastic membrane/Silicon membrane

The diameter of the Silicon membrane was to be greater than 110mm. The silicon will be the main element used for displacing fluid.

It was specified to stretch for a maximum angle of 30° hence the thickness of the silicon rubber was only specified to be 2mm. A suitable material to be used is silicon (medical grade) to meet specified requirements.

Pump top

Material to be used for manufacturing this feature is Teflon. (See material requirements above). The inner diameter of the pump top was also to be 110mm. The inner diameter was chamfered at 45° to reduce turbulence of the fluid that is likely to occur on inner faces of this feature. See **Appendix I** for detailed drawing

Mounting base and rods

The mounting of the pump was designed to be adjustable in order to accommodate various orientations of the pump driver. Mild steel can be used for both features.

Valve design

Proposed design of the valve consisted of the following components: Spring, ball and a spring adjusting bolts. This was only a concept design as proper design to reproduce physiological conditions is a separate challenge as stated in the conclusion of Literature review. Spring and a ball are standard parts and hence not included as parts drawings. These should be made from stainless steel.

The following sketch shows the three component of the valve. The valve is self-centering. Means for adjusting spring force was made by the adjusting bolt. As the bolt is also used preventing water, it is not to be loosened by at least more than half the threaded distance. The valve design would hence require more study.

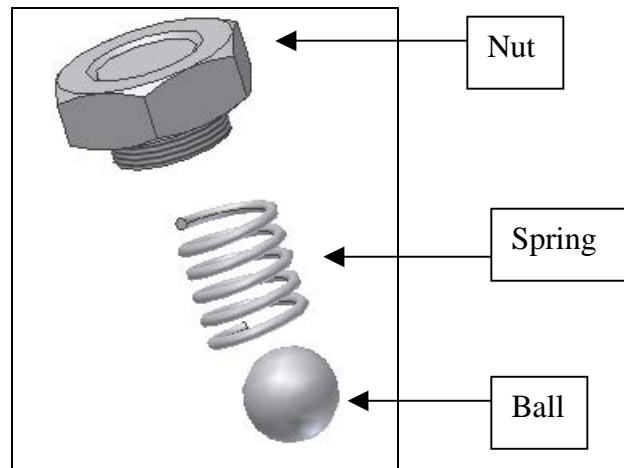


Figure 5-4: Concept of a ball valve with adjustable spring stiffness

Inlet housing and outlet housing

These two components are likely to present manufacturing difficulties. This is due to the holes to be drilled at ninety degrees to each other (See detailed drawing in **Appendix J**). The components could have been made as a solid piece with the pump cover. To minimize manufacturing problems, the two components were designed as separate components and shall be bolted onto the pump base during assembly.

Teflon was chosen as the material for these components.

6 Computer based software development to control the diaphragm pump

As specified in the objectives of the project (section 1.4) a computer based software programme to generate and capture electrical signals was also to be developed.

The reason for the development of software programme was to be able to generate any type of waveforms signals in order to form an input to the pump controller. Once the programme and the pump controller have been incorporated into the Compliance rig, a task which is beyond the scope of the project, generated signals would then be corrected in order to reproduce physiological flow waveforms. The correction of generated signals is another task that is beyond the scope of the project. This task is likely to require knowledge of CONTROL THEORY to accomplish.

Based on the tasks stated above which are outside the scope of the project, it is again reiterated that the tasks that were to be completed in this thesis shall only form the basic starting point from which a complete computer-controlled pump for simulation of cardiovascular conditions can be developed in future.

6.1 Evaluation of computer-based hardware and compatible software to generate and capture signals

Suitable software programs and a computer-based hardware were investigated, as these were not specified at the start of the project. Since there was no allocated budget to purchase software programs and suitable controller during the period of the thesis, readily available devices and software programs had to be used.

It was found that possible computer-based controllers that can be used and that are readily available at the Department of Mechanical Engineering and Electrical Engineering were:

- a. A Data Acquisition Board (**PCI 30GA**, *Eagle Technology*, 16 Channel A/D converter, 4 D/A converter).

The PCI board was found to be readily available in the Mechanical Engineering department. This specific type of data acquisition board is capable of capturing signals via any of the 16 channel analog-to-digital (A/D) converters. It can also produce analogue voltage from any of the 4 digital-to-analog (D/A) channels. The ability of the board to acquire and generate signals makes it suitable for the intended application of the project.

A list of software programs that can be used to interface the board and the PC as listed by the manufacturers were:

LabView, Waveview, Testpoint software. However MATLAB was not listed as one of the suitable programs. The reason for this, as stated by manufacturers was that MATLAB only supported previous board versions before the PCI board. It was later found that MATLAB 6.5 (R 13), the latest version, can be used to interface to any hardware component. [15]

b. Micro-processor, JK1 (MOTOROLA)

This microprocessor is currently used in the Mechanical Engineering and Electrical engineering departments. It doesn't require external software to program it as it consists of built-in software that can be used to generate and capture signals. The device requires prior knowledge of the program used. Once the programming language is understood the device can be programmed to capture and generate signals.

6.1.1 Comparison of PCI30GA board and JK1 microprocessor

In the following table a PCI 30GA (with compatible software) and a microprocessor are compared to determine which device is suitable for the required application of generating and capturing signals to form an input to the pump controller.

For the required application of the thesis, of particular interest were the following factors:

- i. **The ease with which the software can be programmed and used by the end-user.** (Without compromising requirements of the project)

This requirement implies that a software program that is as user-friendly as possible is to be developed.

ii. Resolution / Sampling frequency

Sampling frequency refers to the speed at which the signals are streamed from the device or captured by the device. High sampling frequencies result in high accuracy in the generation or capturing of signals. High resolution also causes high accuracy because resolution refers to the smallest measurable signal by the device.

iii. The ease with which mathematical functions can be manipulated.

As stated in the Literature review that physiological waveforms can be represented by Fourier functions, a software program that can easily manipulate such functions is advantageous.

iv. Cost

The devices to be used belong to the Engineering department hence CVRU Research Unit could require to purchase such devices for own usage in future. Quotations are attached in **Appendix I**.

However for the purpose of the project, availability was a more critical deciding factor than cost.

Table 6-1: Comparison of a PCI 30GA and JK1 Micro-controller

Criteria	PCI 30GA	Micro-processor,JK1
<p>Possible software programs to use</p>	<p>Labview</p> <p>This software is believed to be able to generate any waveform and capture signals. It has been used by Cheng, Jr <i>et al</i> [12] for similar purpose as in this thesis. The software is not readily available in the department and it costs R20 000. A trial version, which is freely available, was found not to have interfaces for the PCI boards. (However normal versions have that capability).</p>	<p>This device can only be programmed its built-in software</p>
	<p>Waveview</p> <p>This software is supplied freely with the board. It is only capable of generating standard waveforms e.g. sine, saw-tooth and square waves and hence is unsuitable for the purpose. However it can be used to monitor and capture signals and it is currently used in the CVRU for a similar purpose.</p>	
	<p>MATLAB</p> <p>Although the suppliers for the board (<i>Eagle Technology</i>) insisted that MATLAB cannot be used to interface with the board, it was specified in the <i>Mathworks website</i> that MATLAB 6.5(R13) can interface with any PCI Board.</p>	
<p>Sampling frequency</p>	<p>The maximum sampling frequency was found to be 100kHz, which is sufficient to accurately sample physiological waveforms.</p>	
<p>Resolution</p>	<p>12 bits</p>	<p>8 bits</p>
<p>User-interface</p>	<p>Through the use of programs like MATLAB or Labview an easy to user interface can be developed to easily generate or capture signals.</p>	<p>The microprocessor only has screen for easy interaction with user. It is only for viewing data.</p>

6.1.2 Justifications for using PCI 30GA board and MATLAB software

Following from the evaluation done on the above table (Table 6.1), PCI Board and *MATLAB 6p5, R13* software were selected. The main advantage of using MATLAB was its ability to manipulate complex functions such as Fourier functions. It was also found that user-interfaces could be designed with MATLAB thus allowing the program to be easily operated by an external user.

The choice of using the PCI board was based on its high resolution and sampling rate compared to the microprocessor. These factors would ensure accuracy in generating and capturing signals. There were other reasons for the choice of a PCI board that were mainly based on its future use at the CVRU (specifically on the Compliance rig). They are as follows:

- i. The device consists of multiple input and output channels and hence various sensors can be interfaced to the card for better control of the generated signal.
- ii. Since the device can generate $\pm 10\text{V}$ and a maximum current of **200mA**. These specifications meet the requirements for an **e/p converter** used to drive the cylinder positioner as this was the method chosen to control the pump (**See chapter 5.2 for the selected pump driver**)

Although the current selling price for the PCI 30GA board is **R6000** (See APPENDIX I for quotations), the advantages listed above justifies the cost of this board.

6.2 MATLAB program developed to generate and capture signals with the PCI30GA board

In this section of the report, the operation of the MATLAB program developed is present and well as features of the developed graphical user-interface.

Shown below is the graphical user interface developed for generating and capturing signals. This system is to be further developed in order to form an input to a full control system of a computer-controller pump for flow simulation.

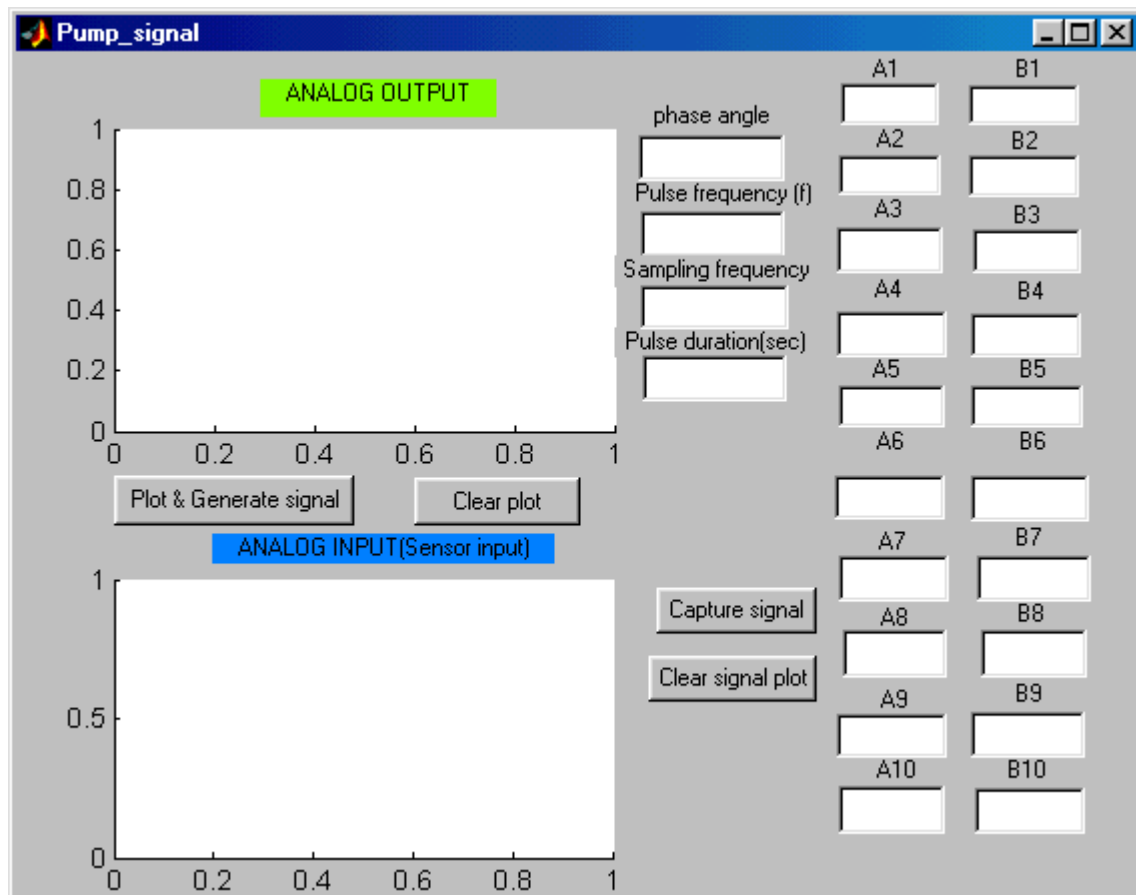


Figure 6-1: A graphical user interface developed for generating waveform signals and capturing of signals

6.2.1 How signals are generated with the developed MATLAB graphical user interface (GUI)

As it was stated that any periodic function could be generated by a Fourier function [4], the main input requirements by the user on the above GUI (figure 6-1) are the Fourier constants represented by **A1 to A10** and **B1 to B10**. The reason for using Fourier constants up to the 10th harmonic was based on the fact that good estimation of physiological waveforms is achieved by Fourier waveforms up to the 6th harmonic [13]. However to accommodate higher accuracy in the simulation of physiological waveforms, the program was designed to generate waveforms up to the 10th harmonic.

Additional inputs required by the user in order to generate waveforms from the PCI card using the developed GUI are:

- **Phase angle:** This parameter indicates the shift of the waveform to be generated from the original zero position. (The parameter must be specified in degrees)
- **Pulse frequency:** The required parameter is related to the heart beat rate at which the pump is to be driven. It must be specified in hertz (Hz) but since the conventional method of representing heart beat rate in cardiovascular terms is in beats per minute (bpm), the relationship between bpm and Hz is (**1Hz = 60 bpm**)
- **Sampling frequency:** This parameter has been explained in Table 6.1. Since the maximum sampling frequency of the board is 100 kHz, it is recommended that this maximum must always be used. The value must be specified in Hertz.
- **Pulse duration:** This implies the period for **reproducing** the signal from the PCI card. The time must be specified in seconds. (It must be noted that, in order to run experiments for very long periods e.g. a very good PC is advantageous (e.g. Pentium 3processors).

Once appropriate Fourier constants and above-mentioned constants have been added on “text boxes”, the function can be plotted and generated by clicking “*Plot and generate button*”. The function to be plotted should appear on the graph space titled: ANALOG OUTPUT.

N.B: If any of the above-mentioned constants are not specified, the waveform shall not be generated.

6.2.2 How signals are captured using the developed GUI

The method used to capture signals is very simple, as it only requires the click of “Capture signal” button. The program that executes in the background handles all parameters that are needed for data to be captured. Refer to MATLAB code in **Appendix G** for parameters required to capture data

For generation and capturing of signals, there is a background code that was developed so that it executes when buttons are clicked after relevant values have been written on relevant text boxes.

Explanation of methods used to develop software program is given in APPENDIX F while a complete code is given in APPENDIX G.

6.3 Apparatus used to test the functionality of the MATLAB program and results obtained

Two devices were used to test the capability of the developed program to generate signal and capture signals.

An oscilloscope: ('LEADER' LBO-524 OSCILLOSCOPE, 35 MHz)

An oscilloscope was used to verify whether the generated signal is similar to the plotted graph titled "ANALOG OUTPUT". To interface the PCI 30GA card to the oscilloscope, a wire was connected from the D/A channel (of the adapter cable) to the input of the oscilloscope. The results obtained shall be displayed below.

A signal generator

This device was to act as a 'dummy' sensor similar to pressure / flow signals that shall be installed on the Compliance rig. The signal generator is capable of generating sine, square or saw-tooth waveforms at a maximum frequency of 1000 kHz

A picture showing the two apparatus i.e. signal generator and oscilloscope can be seen below:

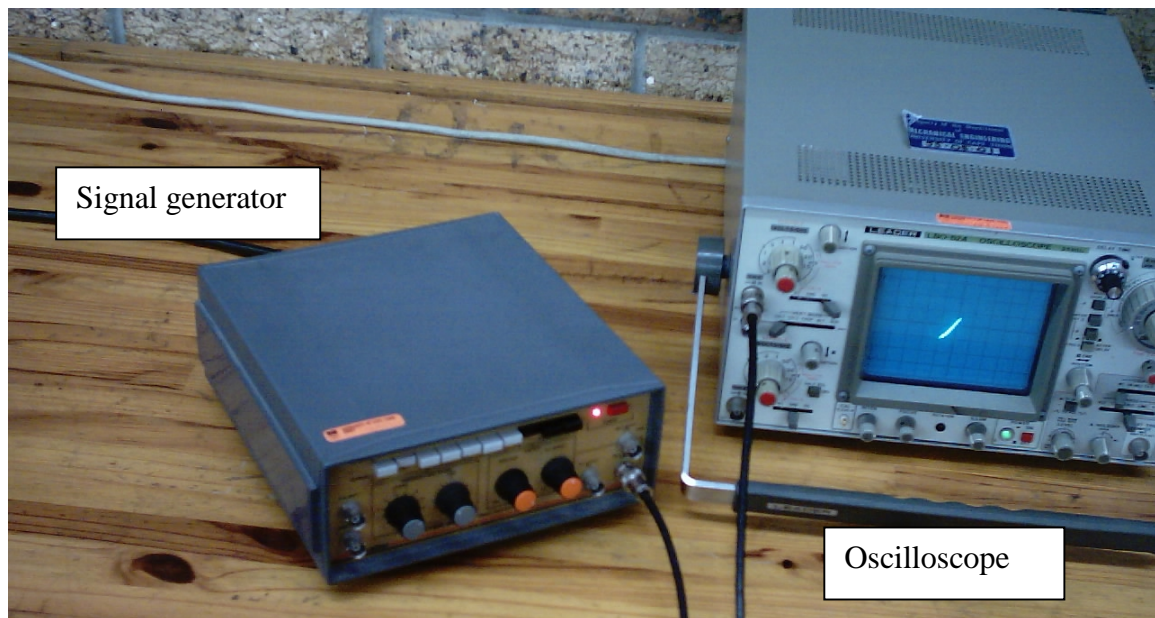


Figure 6-2: Apparatus used to test the functionality of the developed program

Adapter capable to interface a PCI card to the oscilloscope and signal generator

Shown below is a picture of the adapter cable used to interface the PC card to an oscilloscope and signal generator (not shown on the picture below)

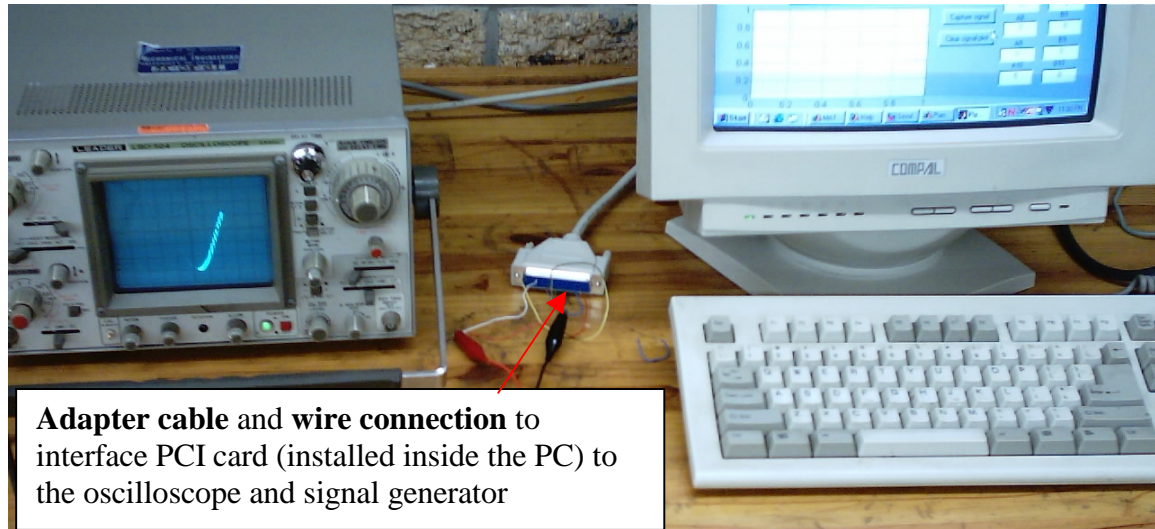


Figure 6-3: A picture showing the adapter from the PC to the signal generator and oscilloscope

Results obtained from the MATLAB program

In this section of the report, results for both generations and capturing of signals are presented. These will be presented simultaneously on the MATLAB GUI developed. It must be noted that the MATLAB program was required to generate any type of waveform, no specific waveform was to be generated. However during the incorporation of the system into the Compliance rig, correct Fourier constants will have to be used in order to reproduce physiologically equivalent waveforms. Correct Fourier constants can be found during the incorporation of the program into the designed pump system in future, as there was not enough time to source this information.

TEST: (Generation of signal at 3Hz and capturing of signal at 100Hz-1.2 kHz).

Generated signal

The signal was generated from required constant shown on the picture below. (Inputs required for signal generation are as explained in **section 6.2.1**)

This signal was verified on the oscilloscope and it was found to correlate with the capture signal on the oscilloscope. (An oscilloscope used didn't have features for storing waveforms hence these could not be presented in the report)

Captured signal

The signal generator was used to generate sine wave. The signal was generated at 100Hz-1.2 kHz. Captured signal can be viewed on the graph entitled ("ANALOG INPUT (Sensor input) on the GUI. Although the basic shape of the signal was captured, the actual values (e.g. amplitude) of the captured signal were not exactly equal to the signal from the signal generator. Based on the shape of the waveform captured, it was concluded that the program functioned as required. It will however require modification.

Calibration of the PCI card, which requires the voltage calibrator, was found to be the course of amplitude mismatch. This device was not present during the execution of the thesis. However the calibration can be done at a later stage.

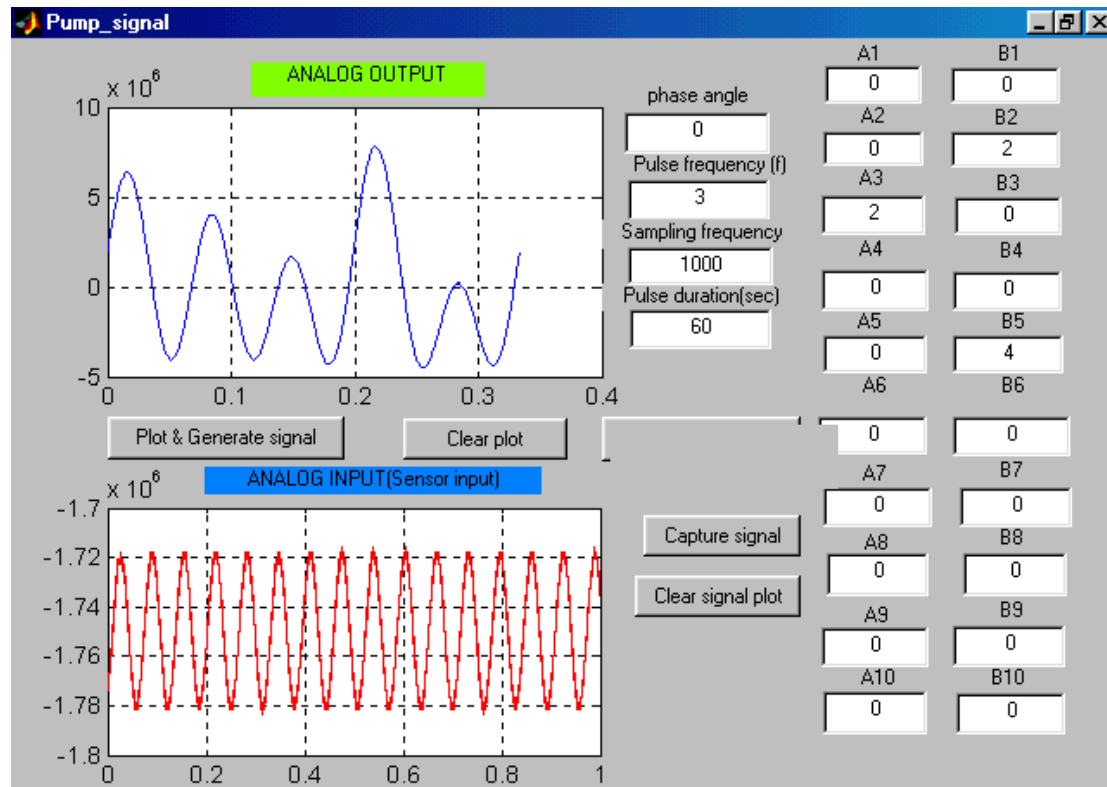
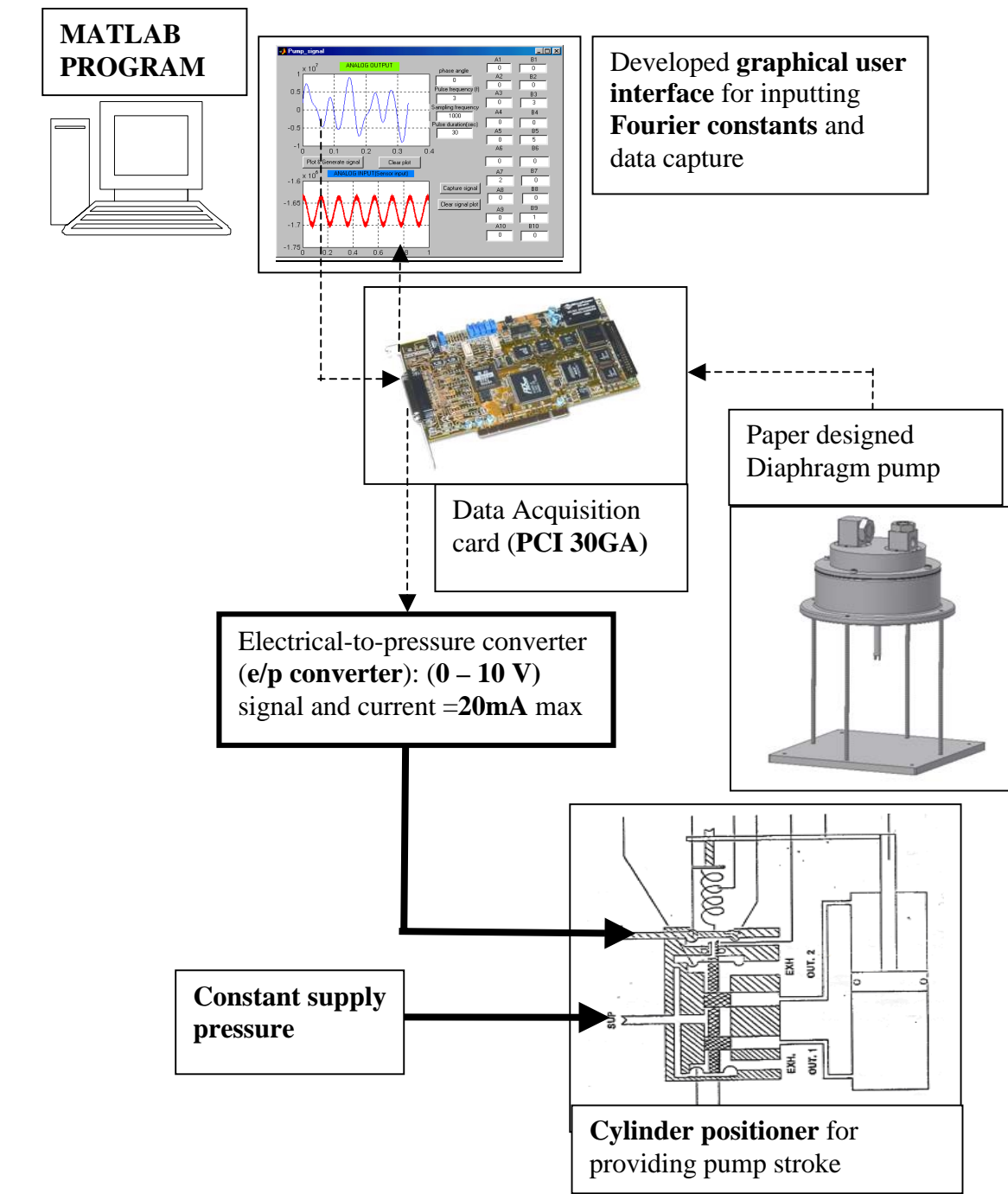


Figure 6-4: Test results for captured signal at 1.2 kHz and generated signal at 3 Hz

For further results of MATLAB program, refer to **APPENDIX H**

6.4 Proposed solution as an integrated system



Key: ----- Voltage signals
 —————> Pressure signals

Figure 6-5: Summary of solution for the design of a computer controlled pump (Not scaled)

6.5 Discussion of the final solutions

Theoretically the proposed system can be used to computer-control the diaphragm pump. Relevant Fourier constants are required as input to the developed graphical user interface.

The PCI card, which can be set to generate signals between 0 – 10V as required by an e/p converter is used to convert program signals into analog voltage signals. The PCI card also produces a maximum current of 200mA hence it meets the requirement of an e/p converter, which requires a maximum of 20mA.

An e/p converter produces pressure, which is proportional to the input voltage signal. The pressure signal, in turn, causes a cylinder positioner that is to be attached to the piston of the diaphragm pump to displace in relation to pressure signals.

Cylinder positioner produces a stroke of 25mm up to 500mm. The diaphragm was designed to achieve a maximum stroke of 25mm with a maximum linear speed of 0.25m/s. The flow-rate of the pump is to be **1 liter per minute**. All the requirements for the pump were verified to comply with cylinder positioner.

This system was however never tested as a unit because of two reasons. The system proposed in the figure above was only theoretically designed (diaphragm pump) and the cylinder positioner was never purchased due budget constraints. Nonetheless the integration of the system was not part of the thesis.

Streaming-out of voltage signals and signal capture was achieved with developed MATLAB program and a PCI 30GA card. An oscilloscope and signal generator tested these two functionalities. The developed program managed to generate any Fourier function up to the 10th harmonic at various pulse frequencies. Data capturing was not satisfactory as it couldn't properly generate signals at higher frequencies > 100kHz

6.6 Expected drawbacks of the system

Possible drawbacks discussed below are expected when the proposed solution is integrated into the Compliance rig. Although the integration of the system into the Compliance rig is beyond the scope of the project, it is a future goal of the Cardiovascular research Unit.

Data Capturing

As discussed above, drawbacks of the system is with data capturing. The program didn't show consistency in capturing data from a signal generator. Other means of data capturing such as *Waveview*, freely available software can be used instead. However this software was not experimented with in the project.

Relating pressure signal of the displaced volume of the pump for control

The proposed solution utilizes pressure to control the position of a cylinder positioner hence the stroke of the pump. This is the method to be used in an attempt to program and hence reproduce physiologically equivalent conditions using a diaphragm pump in a closed loop circuit.

The relationship between the input voltage signal and the displacement of the diaphragm pump has to be accurately determined so that flow-rate can be controlled. In literature, this was determined to the parameter that is to be controlled rather than fluid pressure generated by the pump.

It is envisaged that pressure signal from the e/p converter can tend to overshoot in response to voltage signal hence displacement may not be accurately controlled.

To ensure accurate control of the cylinder positioner stroke even under changing conditions of pressure, feedback loops are normally incorporated on systems. This ensures that input signals are corrected to change output signals so that desired conditions are achieved at all times.

7 CONCLUSIONS

Based on the discussion of the proposed solution in comparison to the objectives set, the following conclusions were drawn. Possible future works to be developed based on the achievements of the projects are also highlighted.

7.1. Diaphragm pumps can be used as a suitable replacement for an existing roller pump

Linear driven pump, of which a diaphragm pump is an example of, were found to be easily controllable as step input signals could be easily related to the pump displaced volume (Volume is directly proportional to displacement). The input stroke of the pump was however not linear in relation to the displaced volume as piston pumps. Diaphragm pump was however chosen due to its advantages to operate without surface-to-surface contacts and thereby reducing leakage problems and maintenance requirements.

It is therefore concluded that the diaphragm pump could be a suitable replacement for a roller pump.

7.2. The software program developed requires improvement for data capturing

A Matlab program that was developed was found to successfully generate any voltage signals up to the 10th harmonics at different pulse frequencies. A similar software program was however not found to capture signals at higher frequencies (100kHz or more). This could compromise the capturing of signals from various sensors on the Compliance rig.

Waveview software, which is freely available with the PCI cards used for programming Fourier functions via MATLAB, is believed to be capable of capturing signals continuously. Since the developed software can capture signals for a specific period of time, Waveview can be used in connection with the developed software to fulfil the requirements of the pump controller to generate and capture voltage signals.

7.3. Feedback loop system can be implemented to improve volume flow rate control of the pump.

It was envisaged that the chosen the pneumatically controlled cylinder positioner would not always accurately control the pump stroke in order to control displaced volume of the pump. It is hence concluded that the implementation of a feedback control system in which flow-rate is measured and correction of an input signal from the PCI card is made until physiologically equivalent flow conditions are achieved.

7.4. Flow sensors, which are not currently installed, are the main requirements implementation of feedback loop.

In order to implement feedback control system flow-rate sensors are necessary for sending feedback signals.

8 RECOMMENDATIONS

Based on conclusions drawn from the project and on the theoretical solutions determined, the following recommendations are made:

9.1. Diaphragm pump should be manufactured once method of control have been further investigated

Detailed drawings of the pumps completed on this project can be used to develop the pump as desired once the cylinder positioner controller has been further investigated as it was only theoretically proven.

9.2. Valve design of the pump should be further investigated in details before the pump can be manufactured

Design of valves was not done in detail in the project hence it requires further investigation.

9.3. Further investigation of control methods of the proposed diaphragm pump including feed-back control methods before implementation of pneumatic control

The method determined for controlling the diaphragm pump should be further investigated as no practical testing of the system was done. The software developed on the thesis only generates various signals that haven't been tested on actuators except oscilloscope.

However, even with the implementation of the proposed cylinder positioner, feedback control methods should be further investigated for possible implementation. Flow rate sensors, not currently installed in the Compliance rig are required for feedback control.

9 REFERENCE

1. Biomedical Science Instrumentation. *Radial Compliance of Natural and Mock Arteries: How this property defines the cyclic loading of deployed vascular stents.* (2000) 38 pp. 163-171. (Retrieved from www.dynatekdelta.com)
2. Charara J, Ruel J, Lemay and C.J. Dillion. *Development of a flow-simulator to study haemodynamic behaviour of natural and artificial blood vessels under the physiological flow conditions.* Journal of Medical Engineering & Technology, Vol. 23 May/June 1999, pg 83-95.
3. KS Vilendrer, TD Nickel and SE Anderson. *Development of a scalable bioreactor for intravascular tissue engineered medical products (ITEMPTS) providing mechanical stimulation and evaluation capabilities.* 2003 Summer Bioengineering Conference (June 25-29). Retrieved from www.tulane.edu/~sbc2003 (2 August 2003)
4. Wilmer W Nichols and Michael F O'Rourke: *McDonald's Blood Flow in Arteries Theoretical, experimental and clinical principle*, 4th edition (Chapter 10)
5. DW Holdsworth, D.W. Rickney, M. Drangova, D.J.M. Miller and A.Fenster. *Computer-controlled positive displacement pump for physiological flow simulation.* Medical & Biological Engineering Computing (1991), Vol. 29 pg 565-570.
6. YF Law, RSC Cobbold, KW Johnston and PAJ Bascom. *Computer controlled pulsatile pump system for physiological flow simulation.* Medical & Biological Engineering Computing (1987), Vol. 25 pg 590-595.
7. Arun KS Iyengar, Hirousatsu, David B Smith and Michael S Sacks. *Dynamic In Vitro Qualification of Bioprosthetic Heart Valve Leaflet Motion Using Structured Light Projection.* Annals of Biomedical Engineering (2001), Vol. 29, pg 963- 973.
8. Millam, R. MSc Thesis. *Design of an adventitial type reinforcement of prosthetic vascular grafts through mechanically affirmed material and structure modulation.* Department of Faculty of Health Sciences, University of Cape Town (2001)
9. Yeoman, M. *Design and optimization of a fabric reinforced porous prosthetic grafts using finite element methods and genetic algorithm*, MSc (2001)
10. Carl. J. Wiggers: *Modern Medical Monographs (Circulatory Dynamics)*, Vol. 4.
11. Arthur M. Brant, Jerome F. Chmielewski, Tin-Kan Hung and Harvey S. Borovetz. *Simulation In Vitro of Pulsatile Vascular Hemodynamics Using a CAD/CAM-Designed Cam Disc and Roller follower.* Artif Organs (1986). Vol. 10 No.5 pg 419-421
12. Rolando Cheng, Jr, Dan L. Ewert and Steven C. Koenig: *Development of a Virtual Instrument for Control and Data Acquisition of a Cardiovascular Pulse Duplicator System Using LabVIEW™.* Retrieved from (http://digital.ni.com/csma.nsf/display?ReadForm&lookup=Biotechnology/Biomedical/Pharmaceutical&view=internetcsbyindustry&node=124200_us) in September 2003

13. Y.C. Fung, *Biomechanics* 4th edition
14. Alberto Giudiceandrea, J Hendryk, Salacinski, Nigel R.M. Tai, Geoffrey Pushon , George Hamilton and Alexander M. Seifalian. *Development and evaluation of an ideal flow circuit: assessing the dynamic behavior of endothelial cell seeded grafts*. Japanese Society for Artificial Organs (2000), pg 17 – 24
15. MathWorks support solution (2003-09-09): Retrieved: September (2003) from www.mathworks.com
16. *Article*: Eric Pepe and George Halfinger. *A New Technology for Metering Fluids in Medical Devices by, KNF Neuberger, Inc* Retrieved from (<http://knf.com/magnetering>)
17. *Article*: Kathleen Berry, Dr. Hans-Joachim Johl and Dr. Eberhard Schluucker *Diaphragm Pumps – An Economic Solution for **Hygienic Application***– The Official Journal of ISPE May/June (2003) Vol. 20 No.3
18. Pediheart organisation (www.pediheart.org) Retrieved September 2003

APPENDICES

APPENDIX A: DIFFERENT TYPES OF PUMPS CONSIDERED

There are two main classes of pumps that can be used for flow simulation and they are Rotodynamic pumps and Positive displacement pumps.

Rotodynamic pumps are classified into a radial pump (centrifugal pump) and axial (propeller pump). These types of pumps transfer energy to fluids via a spinning impeller normally driven by a rotating motor.

Two main classes of positive displacement pumps are Rotary displacement pumps and linear displacement pumps or commonly known as reciprocating pumps. They transport fluid by trapping a fixed volume and then discharge it via gears, diaphragms or pistons.

Research conducted on the types of pumps that are used and have been employed for *physiological flow simulation* indicated a dominant use of positive displacement pumps. However a centrifugal pump was used on one of the flow circuit. A system was reported to produce physiological arterial flow including reverse components but the ease with which they were produced was not disclosed. [14]

It was concluded in the early stages of the project that the rotodynamic pump concepts would not be developed further. A ventricle pump, which doesn't fall under any of the above-mentioned category, has also been used flow simulation by researchers. Its advantages and disadvantages will be explained.

Why Rotodynamic Pumps Are Unsuitable For Physiological Flow Simulation?

Although centrifugal pumps have been used in some medical applications, rotodynamic pumps in general were found to be unsuitable for flow simulation. They have the following disadvantages:

- They are unable to maintain accurate flows under changing inlet and discharge conditions. As accuracy of the pump for flow simulation is important, the use of a rotodynamic pump e.g. centrifugal pump could compromise proper testing of vascular grafts [16].
- They are generally more complex in the design e.g. impeller design.

Possible pump options to replace a roller pump

The following section gives an explanation of different concepts. Each pump concept will be sub-divided into general layout and operation of the pump and valve design (where applicable). Different ways of driving the piston for the motor are also discussed under one section. This is due to the fact that most positive displacement pumps utilise pistons therefore ways of driving it can be similar. The advantages and disadvantages of each concept (which are related to the pump application) will be listed.

Piston pumps

A mechanical piston pump has been reported by KIYOSE *et al* (1977). The operation of these types of pumps is made possible by the fact that any waveform can be represented by its harmonic components [2].

Another design that was reported by WERNECK *et al* (1984) was a servo-driven piston pump. The main disadvantage of this system is that its output waveform was easily interrupted. It was however able to produce ideal flow [5].

Cam-driven piston pumps have been reported by among others: KIYOSE (1977) and POOTS *et al* (1986). It was used to mainly simulate peripheral arterial flow. The main disadvantage of these types of pumps is the difficulty of programming since a dedicated cam shall be required for each distinct waveform to be simulated.

General layout and mode of operation

A simple form of a piston pump consists of a piston located inside a closed chamber. When a piston is drawn back, it creates vacuum inside the chamber and a volume of fluid is drawn in. On the forward stroke, the fluid can then be expelled. Two one-way valves, suction and discharge valves are normally located on the casing to allow transfer of fluid into the chamber during a backward stroke and expulsion of fluid during a forward stroke respectively. To illustrate the components described above, the schematic of a typical piston pump is shown on the following sketch.

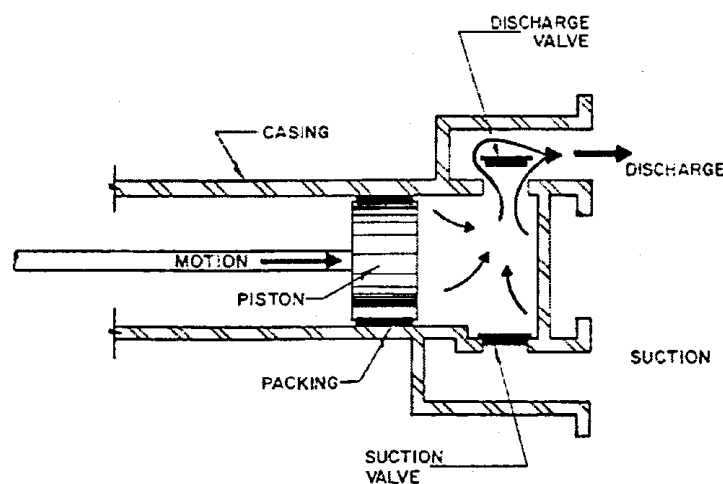


Figure A-1: schematic of a piston pump (KU magazine, www.goae.go.th)

Driving mechanism for piston and diaphragm pumps

In order to drive a piston rod for the above-mentioned pumps, linear motion has to be transferred to the piston rod. Various options available to provide linear motion include: Linear motors, pneumatic cylinders, hydraulic cylinders and conventional rotational motors.

The above options beside rotational motors would readily provide linear motion. If however rotating motors are used, the rotating motion has to be converted to linear motion. The options available to achieve this are:

- i. A rack and pinion arrangement
- ii. Ball screws
- iii. Cam driven by motor and cam follower
- iv. Cog-and-piston rod arrangement (Piston connected to a con-rod)

Valve design for piston and diaphragms

Generally there are two non-return valves required for each type of pump, i.e. suction and discharge valve. The main requirements for the valves are:

- To allow for less flow restriction during each stroke
- Must have a quick response i.e. for closing and opening. A quick response will cause little disturbance of the pressure and flow waveforms being simulated.
- They should be able to provide resisting force that will not cause them to open prematurely.

Advantages of piston pumps are:

- The design of a piston pump can be as simplistic as possible since it uses few parts that are relatively easy to manufacture. (e.g. compared to a diaphragm or centrifugal pump)
- By using a linear motor to drive the piston, the step for the motor can be easily related to the flow-rate. This can simplify the control of the pump to produce pulsatile waveforms, which is a requirement of the project.

Disadvantages of piston pumps are:

- A metal-to-metal contact between the casing and the piston is imminent and wear can occur. If the fluid being pumped is a lubricating type, the friction on the surface of contact can be reduced. However non-lubricating fluids are used for the Compliance rig at CVRU i.e. are water and phosphate buffered saline (PBS) hence contact can cause wear.
- It will be difficult to lubricate the surface of contact since non-lubricating fluid is used.
- The pump seals wear out after undergoing cycles of operation and leaking can occur. Leaking becomes a problem when the piston head has direct contact with the fluid and is used to push the fluid to the discharge point.
- Maintenance requirements of the pump can be high mainly due to wear.

Diaphragm

General layout

These types of pumps normally consist of an elastic material clamped between the piston head and pump housing. The arrangement creates a leak-free environment between the pump chamber and a piston head. Compared to a piston pump, the function of the piston is replaced flexible diaphragm.

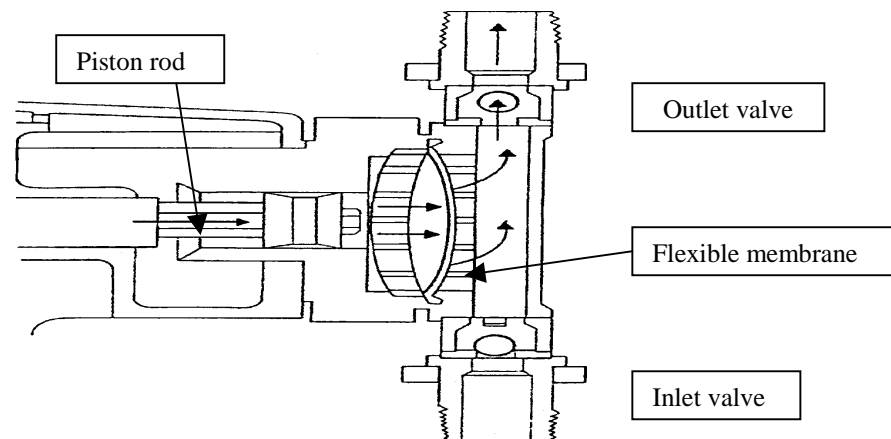


Figure A-2: Diaphragm pump (Picture courtesy of KU magazine)

Advantages

- The use of diaphragm pumps avoids the use of dynamic fluid seals found in piston pumps.
- Since there are no dynamic seals and no metal-to-metal contact of the piston and casing, leakage of the fluid is eliminated.
- The use of elastic material to discharge fluid allow for a gentle fluid outflow.
- They can be designed to avoid metal-to-metal contact and thus allow for lower maintenance

Disadvantages

- Comparatively, it is also the most expensive positive displacement pumps
- Although clamping of diaphragm to piston eliminate the use of dynamic seals, it induces stress concentration area for internal pressures to cause failure [17]

Ventricle pumps

This type of pump was developed by Scima *et al* (1987) [2, 8]. It was reported that the system is only capable of generating flow waveforms not pressure waveforms. Furthermore it can only reproduce human femora artery waveform excellently. It can therefore be concluded that it lacks flexibility to reproduce a variety of waveforms

General layout

A ventricle pump consists of a membrane, which separates the fluid used in the flow circuit and the fluid that provides the required pulses.

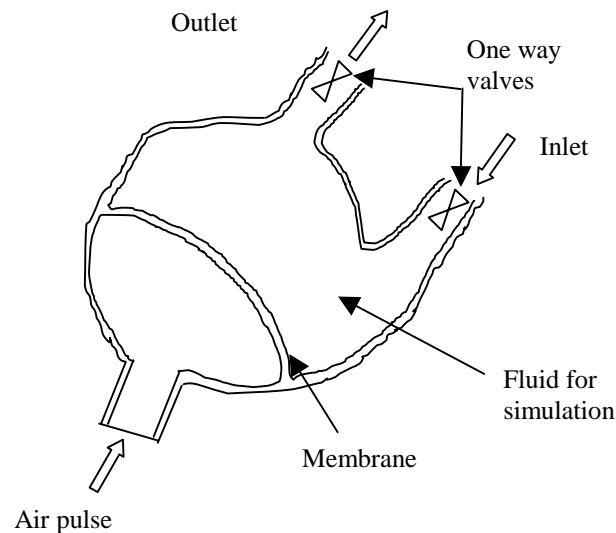


Figure A-3: Schematic of a ventricle pump [8]

The membrane used to separate the air pulse and the fluid used has to be flexible as possible to ensure transfer of pulses.

Advantages:

- This pump system closely mimics the shape and construction of the heart.
- It was found to be able to produce good flow waveforms [8].

Disadvantages:

- It was found that although the pump produces flow waveforms, it is difficult to produce pressure waveforms [8].
- Through previous research, it was also found that the pump could not easily duplicate every physiological waveform [2].
- Since it is not easy to relate the input (air pulse) to the output (flow circuit fluid), the control and hence reproduction of physiological waveforms is very difficult.
- Adequate height is required to create necessary head to ensure proper closing and opening of valves [8].

Driving mechanism

Since air creates quicker pulses than other fluids e.g. water, it can be used for providing pulses to the ventricle pump.

A pneumatic pump will be required to such pulses hence compressed air adds another requirement for the driving mechanism.

Valve design

Fluttering valves can be used as inlet and outlet valves. The fluttering of these valves can cause disturbances of the flow. Flap type valves can be used to eliminate fluttering.

Gear pumps/ Lobe pumps

These are rotary positive displacement pumps that operate by trapping fluid between gears / lobes and the housing and transfer it from the inlet to the outlet of the pump.

Researchers who used these types of pumps are reported by Holdsworth *et al* [5] as ISSARTER *et al* (1978), PETERSEN, (1984), HOSKINS *et al* (1989) among others. One of the main reported disadvantages of gear pumps are that the action of gears can damage the fluid particles under consideration and hence the haemodynamic characteristics can be affected [5 and 6]. They also require lubricant fluid between meshing gears but the liquid used for physiological flow simulation is non-lubricating. Due to damaged haemodynamic characteristics, faulty results can be attained from experiments carried out.

Issarter's design in particular was computer-controlled by servo system in a closed loop Gear pumps require a close clearance between the pump casing and the teeth in order to maintain accurate pumping [4]. Lobe pumps operate in a similar fashion except that they replace gears with much smoother lobes. See figure A-4 and A-5 for the basic concepts of gear and lobe pumps.

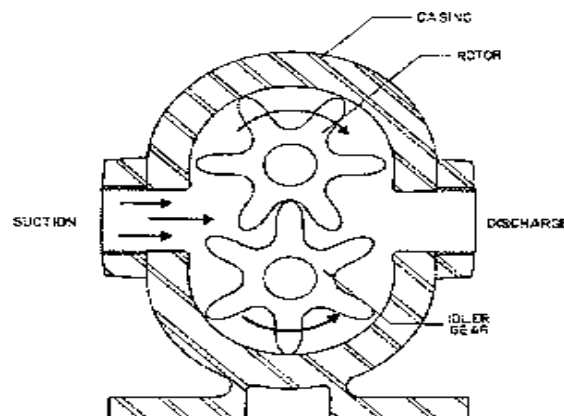


Figure A-4: Gear pump (Picture courtesy of KU magazine)

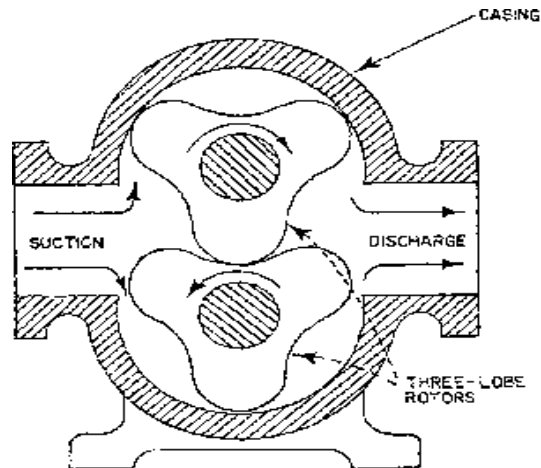


Figure A-5: Lobe pump (Picture courtesy of KU magazine)

Driving mechanisms

The two types of pumps utilise rotary motors to drive them. By controlling the speed of the motor can, the output flow of these pumps can be accurately controlled.

Advantages of gear/lobe pumps

- The main advantage of these types of pumps is that they do not utilise valves. This simplifies the design and flow is not impeded.
- A direct control of the motor connected to one gear can accurately control flow.

Disadvantages

- Wear in these type especially gear pumps, is likely to be more severe than any other types of pumps [5].
- Undesirable noise can result if gears do not properly mesh.
- Since the fluid used for flow circuit simulation is non-lubricating, gears will not be properly lubricated and excessive wear can occur.

APPENDIX B: ANALYSES OF VARIOUS PUMP DRIVES

The controller for the pump plays a significant role in flow simulation. In this section Different pump controllers are analysed.

However from a list of possible drivers stated in Table 2-1 (Literature review) only the cam-driven piston, rack-and-pinion and pneumatic cylinder positioner were ranked. Other systems that were listed but not included in the analysis for reasons given below:

- Linear motors: - Although linear motors can provide displacements that are directly proportional to displaced fluid flow, their cost is higher than conventional rotary motors.
- Ball screws:- The amount of slip/backlash that can occur on these types of linear actuators can result inaccuracy control.

Rack and pinion

A rack is driven by a rotating motor that in turn drives a pinion to produce linear motion in order drive the piston. The teeth of both the rack and pinion must be accurately machined for precise control of the piston movement. Motors required for this application must programmable. Motors that can be used are **stepper motors**.

Advantages of this system are that a relationship between the step input and flow-rate can be formulated for programming of flow waveforms as with Holdsworth *et al*'s design. This will hence not require feedback loop to correct the measured signal if it is found to deviate from the input signal. [5].

The disadvantage with rack-and-pinion systems is that due to high frequencies at which the pump is meant to operate (**3Hz**), noise due to frictional meshing of gears can be undesirable.

Cam-driven piston

The schematic drawing to illustrate the cam-driven piston is shown below:

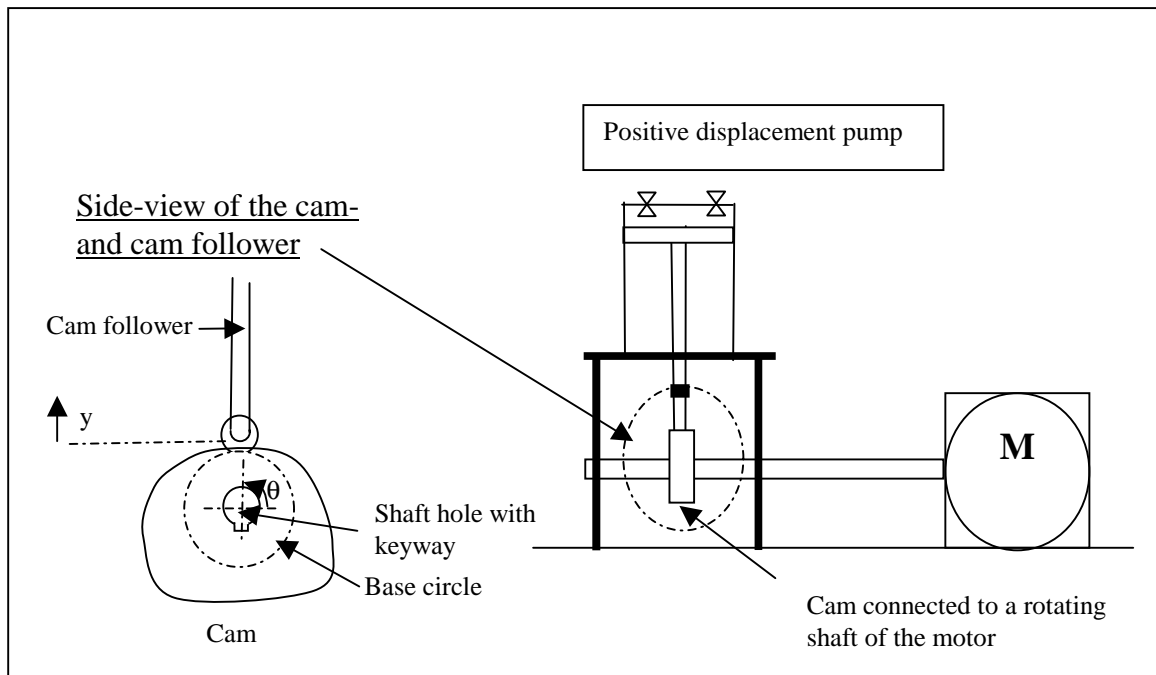


Figure B-1: A sketch showing cam-driven piston pump [11].

The use of a cam-driven piston is not a novel idea in the generation of physiological waveforms. Brant *et al* (1986) have used a similar method and it is reported to produce pressure waveforms in the carotid artery of dogs. For control purposes the rotation of the cam (θ) can be related to the displacement of the piston (y), $y = f(\theta)$.

The use of cams however requires a Computer Aided design and Computer Aided manufacture for a Cam [11]. In order to be able to produce a variety of waveforms, different types of cams will be required for each profile. A lack of flexibility (requirements for different cams) is a major disadvantage of the cam-operated piston.

Crank-driven piston

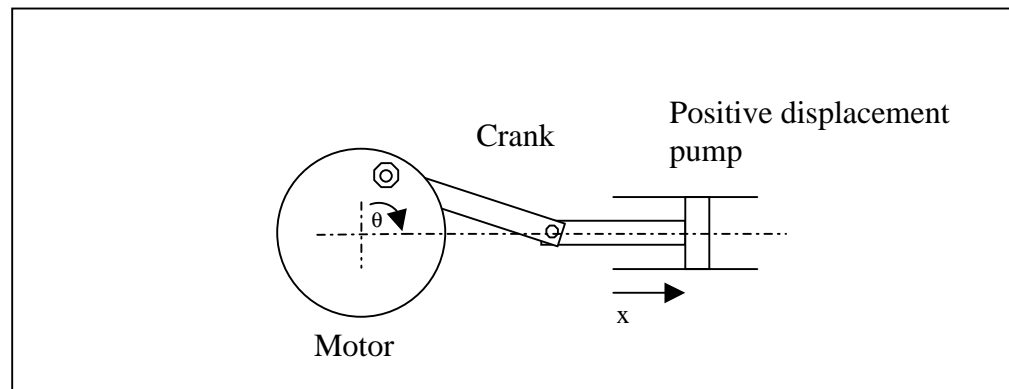


Figure B-2: A sketch for a crank-driven positive displacement pump

A rotating type of motor can be used to drive the crank, which in turn drives the piston on a horizontal axis to provide the required stroke length and hence stroke volume.

One of the main disadvantages of the system is that due to the crank connected between the motor and the piston, the relationship between the rotational input (θ) and the desired output stroke (x) hence flow becomes non-linear. The non-linearity presents difficulty in accurately controlling the output flow-rate.

Pneumatic drives

As it was established that there is compressed air installed in the CVRU laboratory where the Compliance rig is housed, possible ways of controlling the pump with pneumatic air system were investigated.

The other main reason for the investigation of pneumatic drives is the ability of compressed air to quickly respond to an input pressure signal. As quick response was established to be one of the main requirements of suitable pump drive, there was further motivation for the investigation of such systems.

Following from market research, two systems were identified that could be able to pneumatically control the pump. The two systems are:

- a. Proportional pressure regulator and
- b. Cylinder positioner / Proportional positioner

Proportional pressure regulator

A proportional pressure regulator is a device that will output pressure in proportion to electrical input signal (e.g. voltage or current). However the drawback with this method is that it is generally more difficult to control pressure in order to control flow-rate of the pump because pressure depends on load and impedance of the fluid circuit. An easier way is to control stroke of the pump and thereby control flow-rate.

Cylinder positioner

The positioner provides accurate and stable positioning of air cylinders. The system consists of two main components: A voltage-to-pressure converter (e/p converter) and a mechanical system that converts pressure signal from the e/p converter into a cylinder rod movement. See Appendix D for complete specifications of the Cylinder positioner.

Refer to Appendix I for full specifications of a pressure regulator and cylinder positioner.

APPENDIX C: PUMP CALCULATIONS AND EXCEL SHEET

APPENDIX C1: CALCULATIONS AND OPTIMISATION FOR A DIAPHRAGM PUMP

This appendix explains the method used to optimize parameters for a diaphragm pump.

In order to determine characteristics of a diaphragm pump the following conceptual diagram is used: **The requirements for the pump are as according to section 5.2.1 on the main report.**

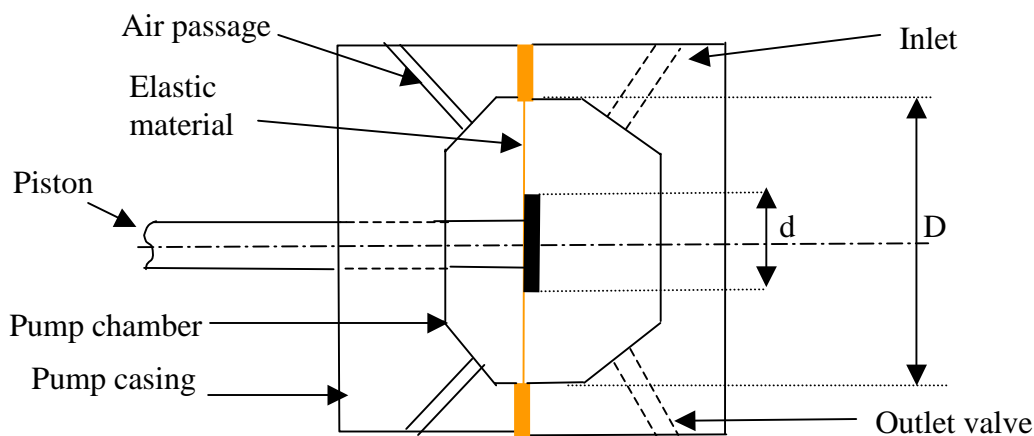


Figure C-1: A schematic diagram for a diaphragm pump concept

The above figure illustrates a concept of the diaphragm pump used to optimise pump parameters. The elastic material is attached at the center by a piston rod that is driven an actuator. Two possible methods of operating the pump were identified. One method is to pull and push the elastic material to both extreme ends. (fig.C-1a). One other method could be to pull it to one end and only allow the piston to stop at the relaxed position of the elastic (fig.C-1b). This method would stop the elastic material from stretching it to other extreme end.

The two options are shown below: (Each figure shows a pump chamber, piston rod and an elastic material):

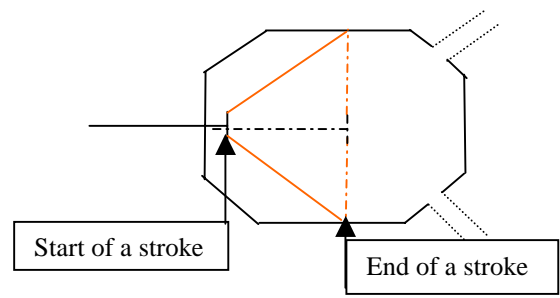
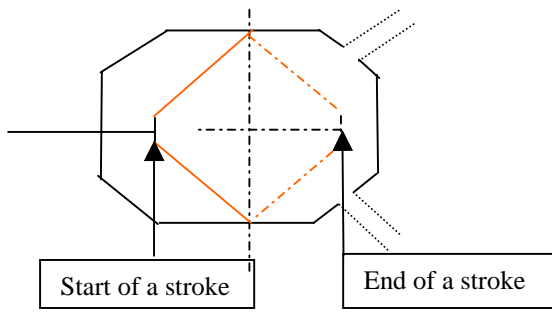


Figure C-2a Stroke length using both extreme ends Figure C-2b Stroke length using one extreme end

The mode of operation shown in FigC-2a causes an elastic material to stretch more than that shown in figure C.2b. A repeated stretching of the elastic material will cause rupture and leakage of fluid can occur. However the idea shown on figure C-2a will ensure more stroke volume to be displaced for similar dimensions in figure C-2b. It must be noted that for both methods, the piston and hence elastic material must first be pulled back to allow fluid inside the chamber via an inlet valve. The forward stroke will then allow fluid to flow out through an outlet valve. The mode of operation shown in figure C-2a will be used for determining equations used for optimising the diaphragm pump. This method is chosen because of minimum strain that it can cause on the elastic material of the pump.

The following figures (fig.C-3a and C-3b) were used to determine the equations for the optimisation of a diaphragm pump.

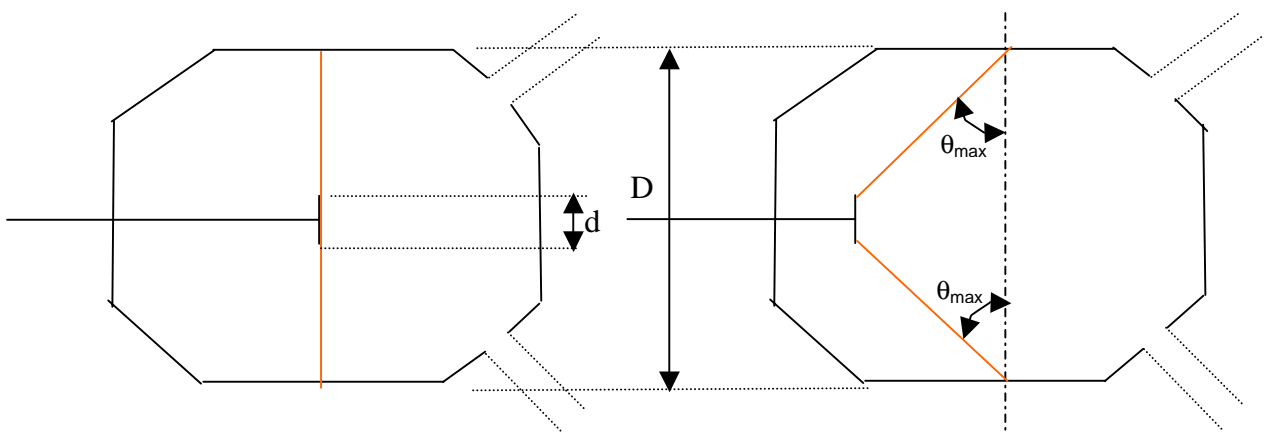


Figure C-3a: The relaxed state of the elastic material retracted state of the pump

Figure C-3b: The

As previously mentioned, during the retraction of the piston rod (hence the elastic material on which it is attached), fluid is allowed in through the inlet valve. At full piston retraction, there is a corresponding maximum angle that the elastic material makes with respect to the vertical as shown in Figure C-3b above. This angle is to be as small as possible since the bigger angle will cause more strain on the elastic material as the number of operating cycles increase hence premature failure can occur.

Figure C-3a also shows the extended state of the elastic material during which the stroke volume of **100ml** of fluid is expelled from the chamber through the outlet valve.

The total volume displaced by the diaphragm pump is therefore represented by the volume swept from the fully retracted state of the elastic material to its fully extended state. This concept is illustrated in figure C-3a and C-3b. The equation used to calculate swept-volume / stroke volume is shown below:

The assumption made to calculate stroke volume was that the elastic material stretched linearly hence the stroke volume will be equal to a **cone**.

$$\text{Since Volume of cone (V}_{\text{cone}}) = \frac{\pi \phi^2 h}{12} \dots\dots\dots(\text{eq.1})$$

where ϕ and h represent the diameter and height of the cone respectively

$$\therefore \text{Stroke Volume (S.V)} = \frac{\pi}{12} \times S.L \left(D^2 + (D^2 - d^2) \frac{d}{D - d} \right)$$

$$\therefore S.V = \frac{\pi}{12} \times S.L (D^2 + Dd + d^2) \dots\dots\dots(\text{eq.2})$$

Optimisation of the stroke length

The stroke length implies the distance travelled by the piston from its full retracted state to its fully extended state. (See figure C-1a and C-1b above). To illustrate the method used to calculate stroke volume, a diagram of the inside chamber of the pump is shown below.

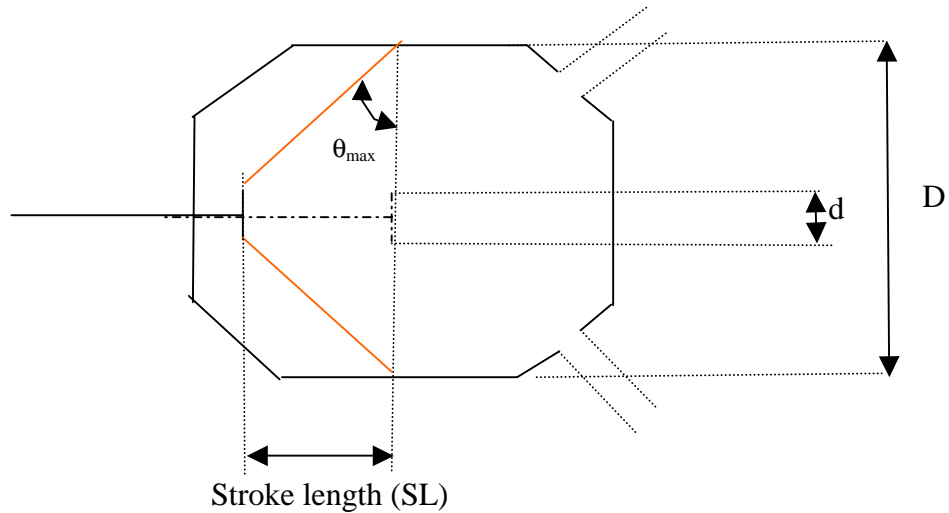


Figure C-4: Illustration of the stroke length

From equation 2 the stroke length can be calculated as follows:

$$\text{Stroke length}(S.L) = \frac{12 * S.V}{\pi(D^2 + Dd + d^2)} \dots\dots\dots eq.3$$

Verification of the flow-rate

The Stroke volume calculated above was used to verify the flow rate. The total volume of 100ml is to be displaced in a systolic time of 0.1 seconds as according specifications in **section 5.2.1** in the main report. The flow rate is therefore:

$$\text{Flow rate}(Q) = \frac{\text{Stroke volume (S.V)}}{\text{systolic time}} \dots\dots\dots eq.4$$

Piston velocity

$$\text{Velocity}(v) = \frac{\text{Stroke volume}}{\text{Systolic time}} \dots\dots\dots eq.5$$

Power required to provide requires fluid displacement

Power (P) = Flow rate × (Maximum Systolic pressure - Minimum Diastolic pressure).

$$P = Q \times (p_{\max} - p_{\min}) \dots\dots\dots eq.6$$

Design criteria used select the diaphragm pump parameters

The selection of parameters of the diaphragm pump was based on the specifications of the cylinder positioner that was chosen as the pump driver.

The specifications for the cylinder-positioner are:

- Maximum stroke = 500mm
- Minimum stroke = 25mm
- Minimum piston diameter = 32mm

The stroke length of the pump is desired to be as short as possible to achieve fast velocity. Hence the optimised stroke was to be in the vicinity of 25mm but not less.

The steps used to select the appropriate stroke-length; maximum angle of the elastic membrane, small diameter (d) and D of the diaphragm pump are outlined below:

- The equation for the stroke volume (equation 1) was set to 100ml as according to the requirement in **section 5-2-1**.
- An Excel tool: Goal Seek was used to set the Stroke length to be **25mm** by setting $d = 25\text{mm}$ and by varying D . The appropriate D was found to be about **110mm**.
- At the same time the angle was to be less than **40 degrees** in order to ensure minimum strain on the elastic material as possible. The calculated angle was optimised to be **30 degrees**.

APPENDIX C2: EXCEL SPREADSHEET FOR OPTIMISATION OF DIAPHRAGM PUMP

Presented on this appendix is an Excel spreadsheet for optimisation of parameters for the piston pump (sheet 1) and diaphragm pump (Sheet 2).

From sheet 1: Piston pump of diameter 70mm or more was found to be suitable for interfacing to the piston pump due to cylinder positioner.

From sheet 2: An *Excel spreadsheet* tool, goal seek, was used to optimize the outer diameter of the diaphragm pump

The main assumption used to determine diaphragm pump parameters was to estimate the stroke volume of the diaphragm pump to the volume of a cone as explained in Appendix C1.

The main criterion used to optimize diaphragm pump calculations was to minimize the stroke length of the pump. This was to ensure maximum velocity of the piston, a requirement which was specified in the main report. The minimum specified stroke length of the cylinder positioner is **25mm** hence this was used to calculate other parameters of the pump such as maximum angle, inner and outer diameter, etc.

The calculations for a conventional piston pump are presented in sheet 1. Since a diaphragm pump is a special type of a piston pump, calculations made in sheet 1 were to identify expected power, torque and force as the last two were not easy to calculate with diaphragm optimisation. Since it involves simpler calculations, it was used to identify how physiological parameters chosen affect the pump parameters such as its power output, force requirement and velocity. The parameters of the piston pump that could meet the requirements of a cylinder positioner are highlighted in sheet 1.

EXCEL SHEET 1: CALCULATIONS FOR A PISTON PUMP

(N.B: Optimum parameters are highlighted in green)

Pump parameters		Physiological parameters	
Stroke volume (SV) Max.	0.0001 m ³	Cycle period (t)	0.333333333 s
Pressure(p _{max})	33.33067 kPa	Systolic (t _s)	0.1 s
Min. Pressure (p _{min})	0 kPa	Diastolic(t _d)	0.233333333 s
		Max. speed(ω)	18.84955592 rad/s
		Tube area (At)	0.000176715 m ²
		Graft area (Ag)	1.25664E-05 m ²
		Flow rate	0.001 m ³ /s

Calculations		Motor parameters						
Piston diameter(D)	Piston area(A)	Stroke length (SL)	Velocity(v _{max})	Flow rate (Q)	Power (P)	Force (F)	Torque(Nm)	Tube velocity (v _t)
0.01 m	7.85E-05 m ²	1.273239545	12.73239545 m/s	0.001 m ³ /s	0.033 kW	0.0026 kN	1.77	5.66 m/s
0.015	0.000177	0.565884242	5.658842421	0.001	0.033	0.0059	1.77	5.66
0.016	0.000201	0.497359197	4.973591972	0.001	0.033	0.0067	1.77	5.66
0.017	0.000227	0.440567317	4.405673165	0.001	0.033	0.0076	1.77	5.66
0.018	0.000254	0.392975168	3.929751681	0.001	0.033	0.0085	1.77	5.66
0.019	0.000284	0.352697935	3.526979348	0.001	0.033	0.0095	1.77	5.66
0.02	0.000314	0.318309886	3.183098862	0.001	0.033	0.0105	1.77	5.66
0.025	0.000491	0.203718327	2.037183272	0.001	0.033	0.0164	1.77	5.66
0.03	0.000707	0.141471061	1.414710605	0.001	0.033	0.0236	1.77	5.66
0.035	0.000962	0.103937922	1.03937922	0.001	0.033	0.0321	1.77	5.66
0.04	0.001257	0.079577472	0.795774715	0.001	0.033	0.0419	1.77	5.66
0.045	0.00159	0.062876027	0.628760269	0.001	0.033	0.053	1.77	5.66
0.05	0.001963	0.050929582	0.509295818	0.001	0.033	0.0654	1.77	5.66
0.055	0.002376	0.042090563	0.420905635	0.001	0.033	0.0792	1.77	5.66

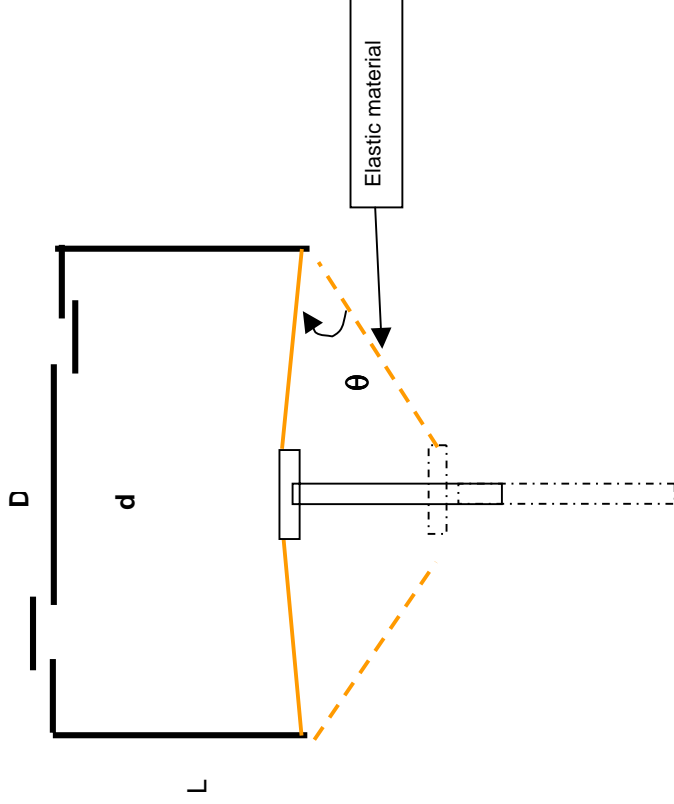
0.06	0.002827	0.035367765	0.353677651	0.001	0.033	0.0942	1.77	5.66
0.065	0.003318	0.030135847	0.301358472	0.001	0.033	0.1106	1.77	5.66
0.07	0.003848	0.025984481	0.259844805	0.001	0.033	0.1283	1.77	5.66
0.075	0.004418	0.02263537	0.226353697	0.001	0.033	0.1473	1.77	5.66
0.08	0.005027	0.019894368	0.198943679	0.001	0.033	0.1675	1.77	5.66
0.085	0.005675	0.017622693	0.176226927	0.001	0.033	0.1891	1.77	5.66
0.09	0.006362	0.015719007	0.157190067	0.001	0.033	0.212	1.77	5.66
0.095	0.007088	0.014107917	0.141079174	0.001	0.033	0.2363	1.77	5.66
0.1	0.007854	0.012732395	0.127323954	0.001	0.033	0.2618	1.77	5.66
0.105	0.008659	0.011548658	0.11548658	0.001	0.033	0.2886	1.77	5.66
0.11	0.009503	0.010522641	0.105226409	0.001	0.033	0.3168	1.77	5.66
0.115	0.010387	0.00962752	0.096275202	0.001	0.033	0.3462	1.77	5.66
0.12	0.01131	0.008841941	0.088419413	0.001	0.033	0.377	1.77	5.66
0.125	0.012272	0.008148733	0.081487331	0.001	0.033	0.409	1.77	5.66
0.13	0.013273	0.007533962	0.075339618	0.001	0.033	0.4424	1.77	5.66
0.135	0.014314	0.006986225	0.069862252	0.001	0.033	0.4771	1.77	5.66
0.14	0.015394	0.00649612	0.064961201	0.001	0.033	0.5131	1.77	5.66
0.145	0.016513	0.006055836	0.060558361	0.001	0.033	0.5504	1.77	5.66
0.15	0.017671	0.005658842	0.056588424	0.001	0.033	0.589	1.77	5.66
0.155	0.018869	0.005299644	0.052996443	0.001	0.033	0.6289	1.77	5.66
0.16	0.020106	0.004973592	0.04973592	0.001	0.033	0.6702	1.77	5.66
0.165	0.021382	0.004676729	0.046767293	0.001	0.033	0.7127	1.77	5.66
0.17	0.022698	0.004405673	0.044056732	0.001	0.033	0.7565	1.77	5.66
0.175	0.024053	0.004157517	0.041575169	0.001	0.033	0.8017	1.77	5.66
0.18	0.025447	0.003929752	0.039297517	0.001	0.033	0.8482	1.77	5.66

EXCEL SHEET 2: DIAPHRAGM PUMP CALCULATIONS
(Physiological conditions)

Stroke volume	0.0001	m ³
Max. Pressure (p _{max})	33.33066688	kPa
Min. Pressure (p _{min})	0	kPa
Systolic time	0.1	s
Diastolic time	0.2333333333	s
Heart rate	0.3333333333	s
Frequency	3	Hz
Tube diameter (d _i)	0.015	m

Optimisation of diaphragm pump parameters
(By choosing D and d and calculating θ and Stroke Length)

D	0.11	m
d	0.025	m
Opt. Stroke length (S.L)	0.02468316	m
θ_{max}	0.52616739	rad
θ_{max}	30.1471706	deg



Calculations

d (m)	0.025	D (m)	0.11	Stroke length (m)	0.024683158	Piston velocity(m/s)	0.246831576	Stroke volume (m ³)	0.0001	Flow-rate (m ³ /s)	0.001	Power (kW)	0.03333067	Tube velocity(v _t)(m/s)	5.65884
-------	-------	-------	------	-------------------	-------------	----------------------	-------------	---------------------------------	--------	-------------------------------	-------	------------	------------	-------------------------------------	---------

APPENDIX D: CYLINDER POSITIONER

The cylinder positioner specified was an “IP-200 Cylinder Positioner Model”

Important specifications for the positioner are:

Supply Pressure: 300-700kPa

Signal Pressure: 20-100kPa

Linearity: Less than $\pm 2\%$

Hysteresis: Less than 1%

Repeatability: Less than 1%

Sensitivity: Less than 1%

Air Consumption: Less than 22 L/min

Maximum Air Flow: 250 L/min (at 500kPa)

Supply pressure variation: 0.5 % (at ± 50 kPa supply)

Applicable Cylinder: 32 mm or larger diameter

Stroke: Minimum = 25mm and Maximum is 500mm

Operating temperature: -5 C to 60C

Response time:

A diagram for the cylinder positioner is given in the figure below:

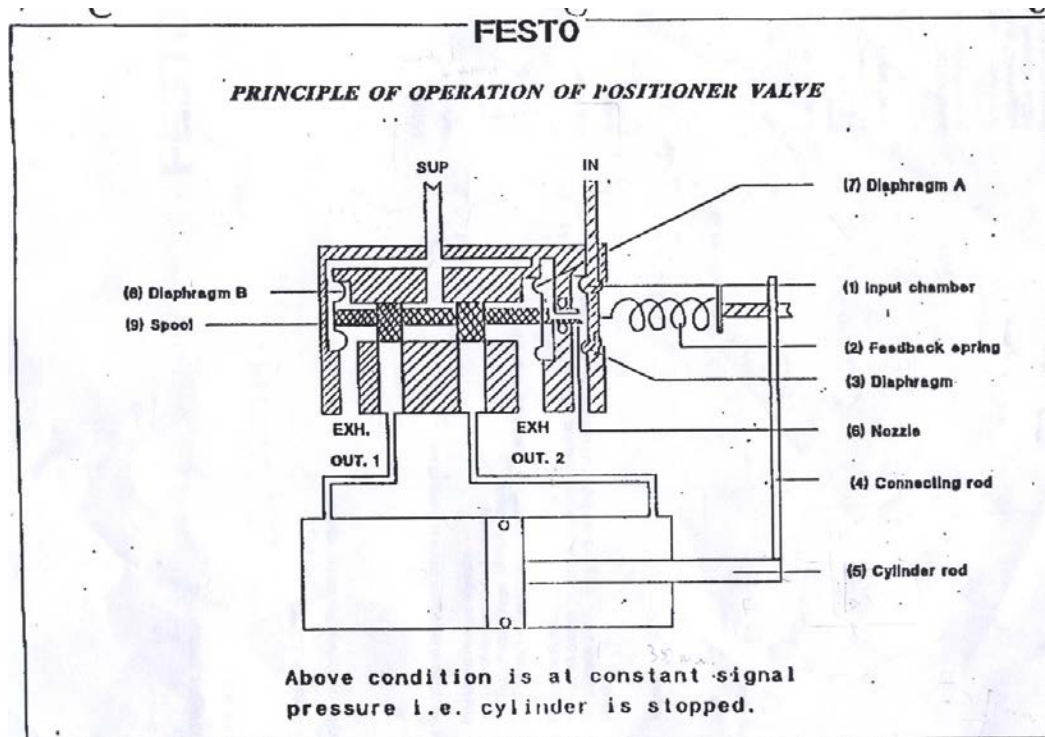


Figure D-1: Principle of operation of position valve (FESTO)

APPENDIX E: VIVITRO SYSTEM

In this appendix a system developed by *Vivitro System Inc.* (Superpump System SPS3891) is presented that is believed to meet the requirements of a computer-controlled positive displacement for testing mechanical properties of blood vessels.

The system as explained below is reported to be capable of producing cardiovascular waveforms. Full explanation of different components of the system is as according to the *Vivitro system website* (<http://www.vivitro.bc.ca>)

GENERATION OF CONTROLLED OSCILLATORY FLUID FLOWS

The SUPERPUMP system consists of a piston-in-cylinder pump head driven by a low inertia electric motor. A linear actuator converts rotary motion of the motor to linear displacement of the piston using a lead screw. A power amplifier drives the motor. Position and velocity transducers provide feedback to the amplifier.

A SUPERPUMP system includes amplifier, linear actuator, motor, pump head, motor cooling fan, and interconnecting cables. Complex fluid flows can be generated by input of an appropriate waveform to the power amplifier.



Figure E-1: Superpump System SPS3891

Vivigen Waveform Generator VG2001

PC CREATION & PLAYBACK of COMPLEX ARBITRARY WAVEFORMS

The VIVIGEN Waveform Generator unit VG2001 can be used stand-alone or can be controlled by a PC through a serial line. In stand-alone mode, 7 arbitrary waveforms can be switch selected for output, each at a set rate. Using a PC, and the included software, up to 7 arbitrary waveforms can be downloaded and each can be set to output at a rate in the range 1- 600 cycles/minute.

The software includes a module for designing waveforms from SIN or multi-Linear functions for driving SUPERPUMP operating in POSITION or VELOCITY SERVO feedback mode. The Waveform Generator output can be synchronised to other VG2001 units. Up to 3 Waveform Generators can be accommodated in a single TRI-PACK chassis.



Figure E-2: Vivigen Waveform Generator VG2001

Viscoelastic Impedance Adapter VIA7991

HYDROMECHANICAL SIMULATION OF VISCOELASTICITY

The VISCOELASTIC IMPEDANCE ADAPTER, VIA7991 interfaces between a fluid flow generator and a load. It consists of a unique combination of resistive and compliance elements which simulate viscoelastic properties. This helps produce

realistic ventricle pressure waveforms in pulse duplicator systems simulating cardiovascular function. In prosthetic heart valve studies ventricle pressures with physiologic dp/dt values can be achieved using the VIA.

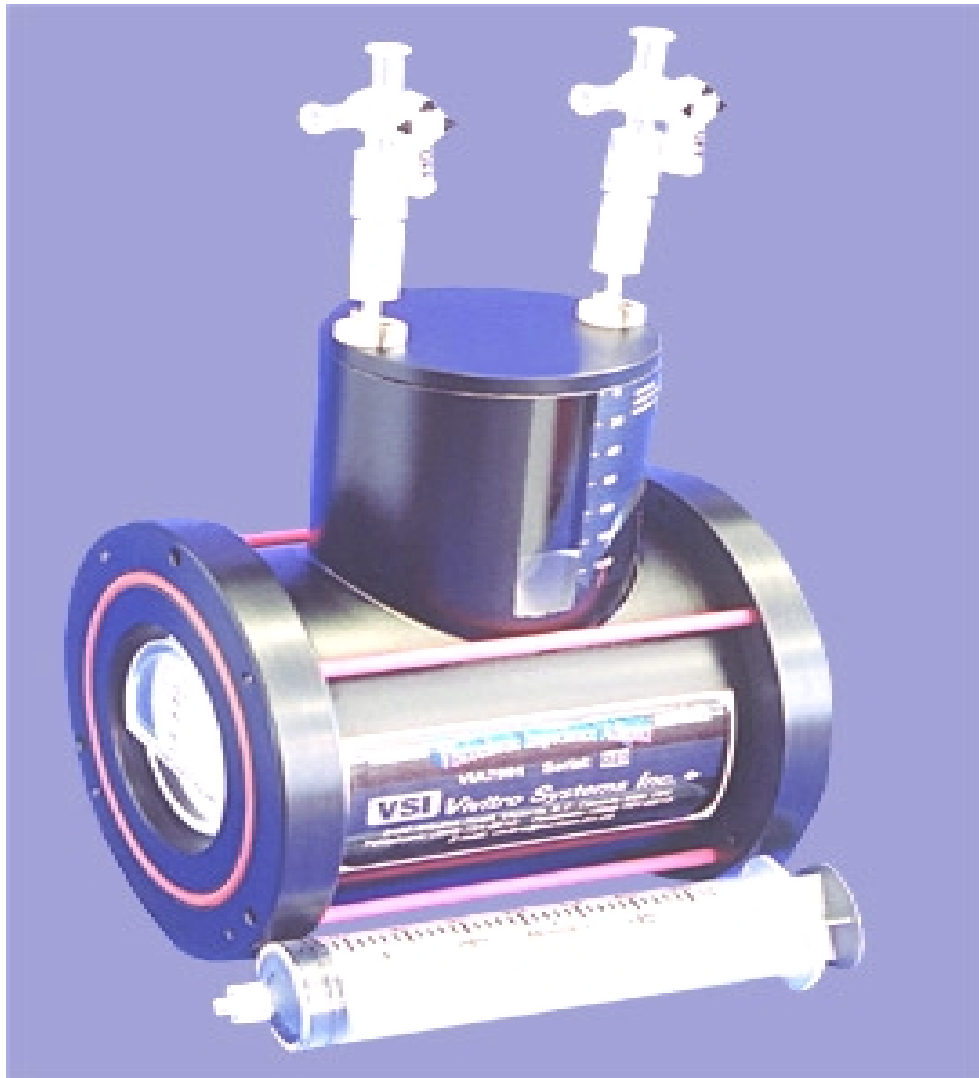


Figure E-3: Viscoelastic Impedance Adapter, Via7991

APPENDIX F: EXPLANATION OF MATLAB CODE

MATLAB PROGRAM USED TO GENERATE SIGNALS VIA PCI CARD AND CAPTURE SIGNALS (See the MATLAB program(Appendix G) attached for further clarification)

Resources used to program the PCI 30GA card

The most important documents used to program functions to and from the PCI card was the PCI30FG User Manual. MATLAB 6.5 version was also used as previous MATLAB versions only supported a limited PCI cards from selected manufacturers. Eagle Technology (South Africa) was one of the manufacturers not supported.

A package that is freely available from the MATHWORKS website, named “generic dll.exe” was required to links functions for the PCI card and the Matlab6-5 so that such functions can be directly called from Matlab6.5.

This package also contained a pdf file named “dll_interface_guide”. This file explained methods used to interface MATLAB to the functions of the PCI card.

Advices from the MATLAB support were given through e-mail as the method communicating with the PCI card was still novel.

Generation of signals via a PCI card

Before signals can be generated, the dynamic link library that contains the functions that can be used to communicate with the card have to be load: The command to be used is “loadlibrary (edrapi)” – edrapi is the name of a library that contains functions for the PCI card.

In order to generate signals from the PCI card using MATLAB: Required functions are:

- EDRE_DAConfig: This function is used for choosing frequency at which data is to be output, channel from which data will be output.

- EDRE_DAControl: This command is used to start the generation of specified voltage function and to eventually stop the process. Time in second can be specified by the user for how long the signal can be generated. The command in MATLAB that carries this out is “pause (s)” where s specifies the time to generate signal.

After the program completes execution, the library must be unloaded with:
“unloadlibrary command”

Signal capture

Similar method for loading and unloading library applies.

Functions required to generate signals are:

- EDRE_ADGetData: This command is used for specifying where the signals captured shall be stored.
- Once the above function is called, the process can be started and stopped again by specifying the time period for data capturing: EDRE_ADStop and EDRE_ADStart are used for stopping and starting respectively.

For further clarification of the program refer to the comments made on the program (APPENDIX G)

Program for Graphical User Interface (GUI)

Edit buttons, text, graphs and launch buttons on the GUI are easily created from the table that gets launched when “guide” command is started from the PC. Once the buttons have been created, M-files are automatically created to link actions of the button with the execution of the program. The created M-file must be edited to operate according to the action of the user. The most important button for the developed are the edit buttons used for accepting input from the user.

APPENDIX G: DEVELOPED MATLAB CODE

```
% MATLAB CODE for generating and capturing signals by I. Pitiri
% for Thesis: "Development of a positive displacement pump"

function varargout = Pump_signal(varargin)
% PUMP_SIGNAL M-file for Pump_signal.fig
% PUMP_SIGNAL, by itself, creates a new PUMP_SIGNAL or raises the existing
% singleton*.
%
% H = PUMP_SIGNAL returns the handle to a new PUMP_SIGNAL or the handle to
% the existing singleton*.
%
% PUMP_SIGNAL('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in PUMP_SIGNAL.M with the given input arguments.
%
% PUMP_SIGNAL('Property','Value',...) creates a new PUMP_SIGNAL or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before Pump_signal_OpeningFunction gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to Pump_signal_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Pump_signal

% Last Modified by GUIDE v2.5 13-Oct-2003 00:52:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Pump_signal_OpeningFcn, ...
    'gui_OutputFcn', @Pump_signal_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Pump_signal is made visible.
function Pump_signal_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to Pump_signal (see VARARGIN)

% Choose default command line output for Pump_signal
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```

% UIWAIT makes Pump_signal wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Pump_signal_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
% str2double(get(hObject,'String')) returns contents of edit2 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit3_Callback(hObject, eventdata, handles)
% hObject   handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%       str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit4_Callback(hObject, eventdata, handles)
% hObject   handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%       str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata, handles)

```



```

% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
% str2double(get(hObject,'String')) returns contents of edit5 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit6_Callback(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit6 as text
% str2double(get(hObject,'String')) returns contents of edit6 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit7_Callback(hObject, eventdata, handles)
% hObject handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit7 as text
% str2double(get(hObject,'String')) returns contents of edit7 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc

```

```

    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit8_Callback(hObject, eventdata, handles)
% hObject   handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%       str2double(get(hObject,'String')) returns contents of edit8 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit9 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit9_Callback(hObject, eventdata, handles)
% hObject   handle to edit9 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%       str2double(get(hObject,'String')) returns contents of edit9 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit10 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit10_Callback(hObject, eventdata, handles)
% hObject   handle to edit10 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%       str2double(get(hObject,'String')) returns contents of edit10 as a double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function edit11_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit11_Callback(hObject, eventdata, handles)
% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
% str2double(get(hObject,'String')) returns contents of edit11 as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit12_Callback(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
% str2double(get(hObject,'String')) returns contents of edit12 as a double

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit13_Callback(hObject, eventdata, handles)
% hObject handle to edit13 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
% str2double(get(hObject,'String')) returns contents of edit13 as a double

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit14_Callback(hObject, eventdata, handles)
% hObject handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
% str2double(get(hObject,'String')) returns contents of edit14 as a double

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit15_Callback(hObject, eventdata, handles)
% hObject handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
% str2double(get(hObject,'String')) returns contents of edit15 as a double

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');

```

```

else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16 as a double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit17_Callback(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%        str2double(get(hObject,'String')) returns contents of edit17 as a double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function edit19_CreateFcn(hObject, eventdata, handles)

```

```

% hObject handle to edit19 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit19_Callback(hObject, eventdata, handles)
% hObject handle to edit19 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
% str2double(get(hObject,'String')) returns contents of edit19 as a double

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit20 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit20_Callback(hObject, eventdata, handles)
% hObject handle to edit20 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
% str2double(get(hObject,'String')) returns contents of edit20 as a double

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit21 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit21_Callback(hObject, eventdata, handles)
% hObject handle to edit21 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit21 as text
% str2double(get(hObject,'String')) returns contents of edit21 as a double

% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit22 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit22_Callback(hObject, eventdata, handles)
% hObject handle to edit22 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit22 as text
% str2double(get(hObject,'String')) returns contents of edit22 as a double

% --- Executes during object creation, after setting all properties.
function edit23_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit23 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit23_Callback(hObject, eventdata, handles)
% hObject handle to edit23 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit23 as text
% str2double(get(hObject,'String')) returns contents of edit23 as a double

% --- Executes on button press in pushbutton1.
%function varargout pushbutton1_Callback(hObject, eventdata, handles, varargin)
function varargout = pushbutton1_Callback(hObject, eventdata, handles, varargin)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%START OF THE CODE TO GENERATE AND CAPTURE SIGNALS ON BUTTON PRESSES

A1 = str2double(get(handles.edit1,'String'));
A2 = str2double(get(handles.edit2,'String'));
A3 = str2double(get(handles.edit3,'String'));

```

```

A4 = str2double(get(handles.edit4,'String'));
A5 = str2double(get(handles.edit5,'String'));
A6 = str2double(get(handles.edit6,'String'));
A7 = str2double(get(handles.edit7,'String'));
A8 = str2double(get(handles.edit8,'String'));
A9 = str2double(get(handles.edit9,'String'));
A10 = str2double(get(handles.edit10,'String'));
B1 = str2double(get(handles.edit11,'String'));
B2 = str2double(get(handles.edit12,'String'));
B3 = str2double(get(handles.edit13,'String'));
B4 = str2double(get(handles.edit14,'String'));
B5 = str2double(get(handles.edit15,'String'));
B6 = str2double(get(handles.edit16,'String'));
B7 = str2double(get(handles.edit17,'String'));
B8 = str2double(get(handles.edit18,'String'));
B9 = str2double(get(handles.edit19,'String'));
B10 = str2double(get(handles.edit20,'String'));
f = str2double(get(handles.edit21,'String'));
phi = str2double(get(handles.edit22,'String'));
s = str2double(get(handles.edit25,'String')); % Length of pulse generation in seconds
sf = str2double(get(handles.edit26,'String')); % Sampling frequency N.B: For PCI30GA maximum is 5kHz
w = 2*pi*f; % Pulse/plot frequency in hertz
t = linspace(0,(1/f),sf); % Time period for plot
F = 1e6*(0.5*A1 + A1.*cos(w.*t-phi) + A2.*cos(2*w.*t-phi) + A3.*cos(3*w.*t-phi)+ A4.*cos(4*w.*t-phi)+
A5.*cos(5*w.*t-phi)+ A6.*cos(6*w.*t-phi)+ A7.*cos(7*w.*t-phi)+ A8.*cos(8*w.*t-phi)+ A9.*cos(9*w.*t-phi)...
+ A10.*cos(10*w.*t-phi)+ B1.*sin(w.*t-phi)+ B2.*sin(2*w.*t-phi)+ B3.*sin(3*w.*t-phi)+ B4.*sin(4*w.*t-phi)+
B5.*sin(5*w.*t-phi)+ B6.*sin(6*w.*t-phi)+ B7.*sin(7*w.*t-phi)+ B8.*sin(8*w.*t-phi)+...
+ B9.*sin(9*w.*t-phi) + B10.*sin(10*w.*t-phi));

axes(handles.axes1)
set(handles.axes1,'XMinorTick','on')
plot(t,F)
grid on
loadlibrary('c:\WINDOWS\system\edrapi.dll','c:\matlab6p5\toolbox\matlab\general\edrapi.h')

libf = libfunctions('edrapi','-full');

% Required constants

Sn = uint32(1000000555); % Board serial number
sf_ = uint32(sf); % Sampling frequency
c = uint32(0); % Dac channel to be used
m = uint32(1); % Loop mode or non-loop mode?

calllib('edrapi','EDRE_DACconfig',Sn,c,sf_,uint32(3),uint32(0),m,length(F),F) % Channel configuration
calllib('edrapi','EDRE_DACControl',Sn,c,uint32(1)) % Start voltage generation

pause (s); % Wait for operator's response

calllib('edrapi','EDRE_DACControl',Sn,c,uint32(2)) % Stop the process
calllib('edrapi','EDRE_DAWrite',Sn,c,uint32(0)) % Reset voltage to zero for next operation
unloadlibrary('edrapi')

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

axes(handles.axes1)
cla

% Add a CODE that links the function F to the PCI30GA card

% --- Executes on button press in pushbutton4.

```



```

function varargout = pushbutton4_Callback(hObject, eventdata, handles, varargin)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function edit25_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit25 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit25_Callback(hObject, eventdata, handles)
% hObject handle to edit25 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit25 as text
% str2double(get(hObject,'String')) returns contents of edit25 as a double

% --- Executes during object creation, after setting all properties.
function edit26_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit26 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit26_Callback(hObject, eventdata, handles)
% hObject handle to edit26 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit26 as text
% str2double(get(hObject,'String')) returns contents of edit26 as a double

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
function varargout = pushbutton5_Callback(hObject, eventdata, handles, varargin)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Scanning ADC subsystems
%clc
%clear all
loadlibrary('c:\WINDOWS\system\edrapi.dll','c:\matlab6p5\toolbox\matlab\general\edrapi.h')
libfu = libfunctions('edrapi','-full')

```

```

% Constants required for inputting signal

Sn = uint32(1000000555);           % Board serial number
sf = uint32(100000);              % Max. Sampling frequency(100kHz)
fp = libpointer('uint32Ptr',sf);  % Frequency pointer
Clk = uint32(1);                  % Internal clock
Burst = uint32(0);                % Burst mode otherwise 0 rep normal mode
R = uint32(0);                    % Voltage range
GainList = uint32([0]);           % Gain of channel 13
G_p = libpointer('uint32Ptr',GainList); % Gainlist pointer
ChanList = uint32([5]);           % Channel(s) to be scanned
C_p = libpointer('uint32Ptr',ChanList); % Channel list pointer
LSize = uint32(1);                % Depth of channel list
pT = 1;                            % Period of plot
Buf = int32(zeros(100000,1));
Bufsize = uint32(length(Buf));
Buf_p = libpointer('int32Ptr',Buf);
Bufsize_p = libpointer('uint32Ptr',Bufsize);
calllib('edrapi','EDRE_ADConfig',Sn,fp,Clk,Burst,R,C_p,G_p,LSize)
calllib('edrapi','EDRE_ADStart',Sn) % Start A/D scanning
pause(5)
[ga gb gc] = calllib('edrapi','EDRE_ADGetData',Sn,Buf_p,Bufsize_p) % Retrieve samples
calllib('edrapi','EDRE_ADStop',Sn) % Stop

calllib('edrapi','EDRE_Query',Sn,uint32(109),uint32(0)) % Query the number of samples available
%Checking the number of samples returned

t = linspace(0,pT,length(gb)); % Plot data retrieved (gb) for a period=pT
axes(handles.axes2)
set(handles.axes2,'XMinorTick','on')
plot(t,gb,'r')
grid on
unloadlibrary('edrapi')

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
axes(handles.axes2)
cla

```

APPENDIX H: RESULTS FOR SIGNAL CAPTURE AND GENERATION OF DEVELOPED PROGRAM

TESTING AT LOWER FREQUENCY

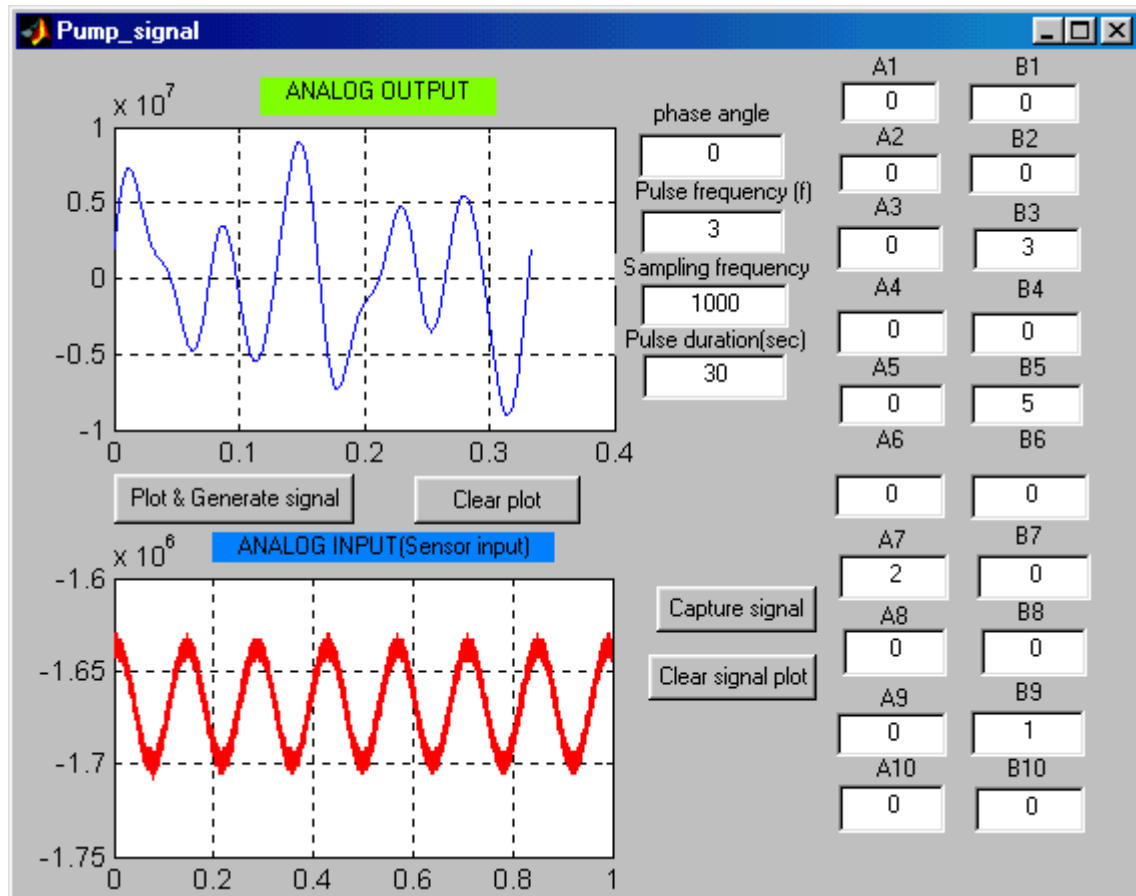


Figure H-1: Signal capture 1-1.2Hz

Generated signal:

This is to show that various waveforms can be generated hence the composition of any signal to form an input to the pump can be achieved.

Captured signals

Testing at lower frequencies: 1-1.2Hz @ frequency multiplier: coarse: 10, fine: -0.8

ESTING AT SIGNAL FREQUENCY: 1-12Hz

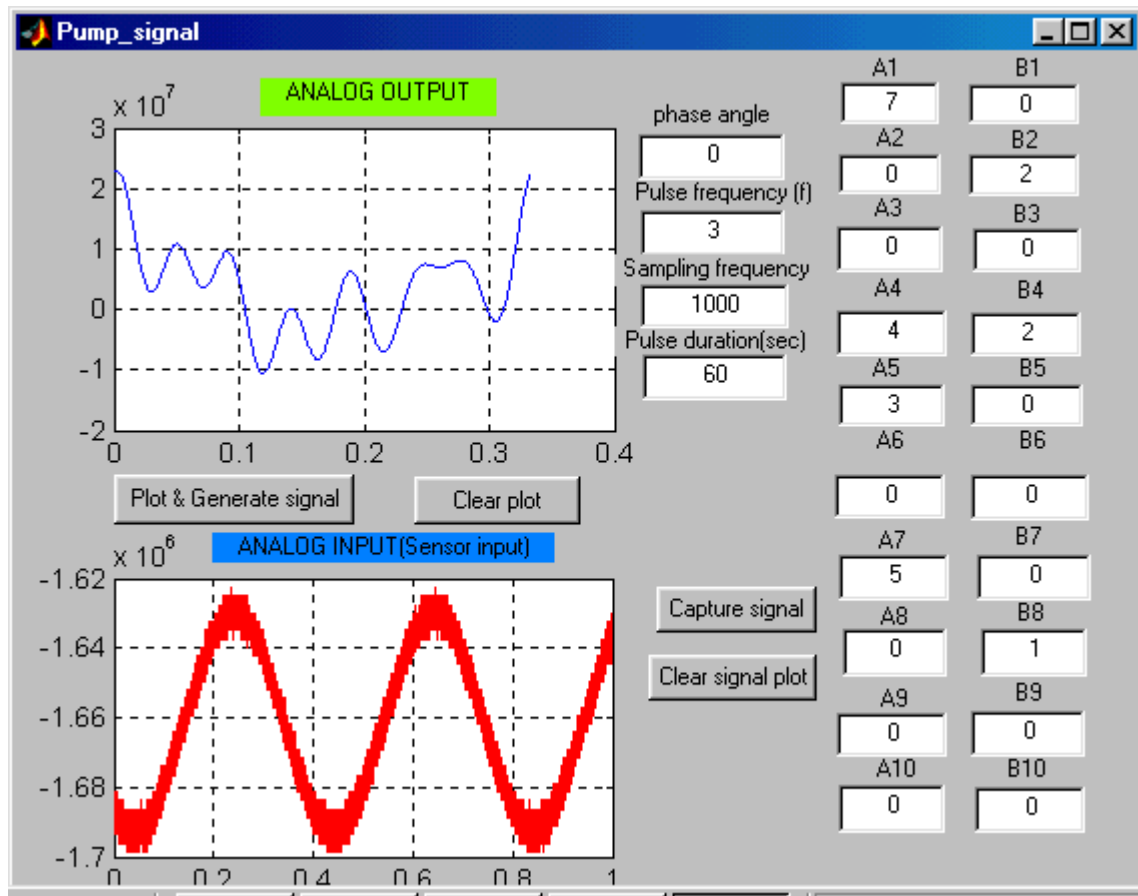


Figure H-2: Signal capture at 12 Hz

CAPTURING A SQUARE WAVE SIGNAL

An example of a square wave: @1 – 12Hz

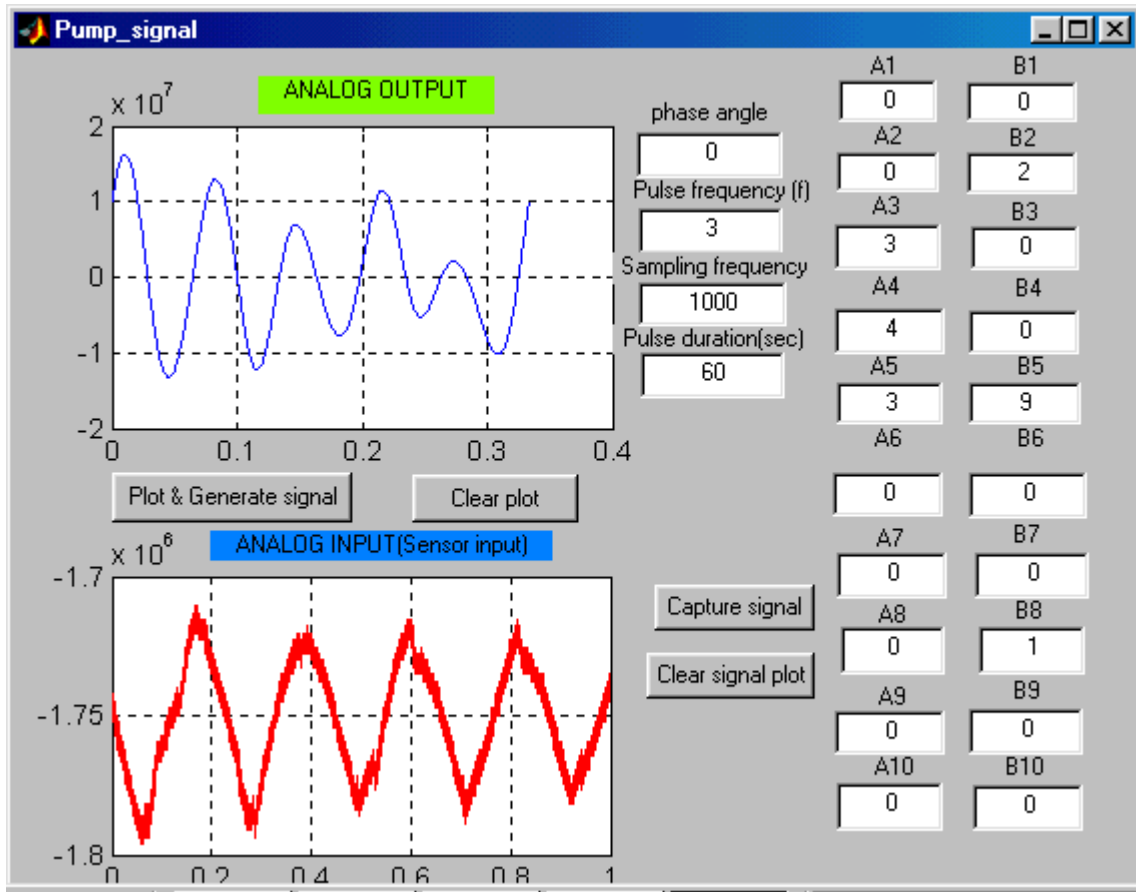


Figure H-3: Signal capture of a square wave

SIGNAL CAPTURE AT 100kHz

At 100 kHz, the signal was not easily visible (See graph for analogue output on the GUI)

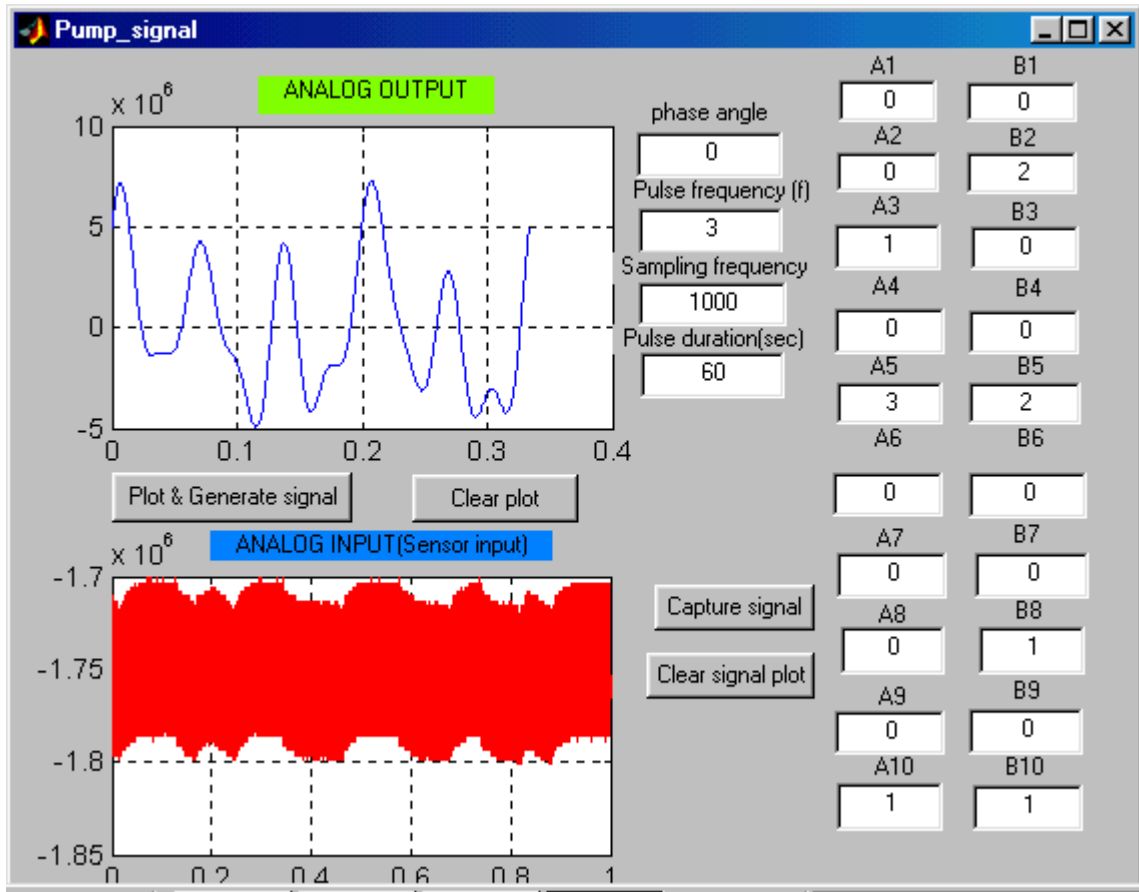


Figure H-4: Signal capture of a sine signal at 100kHz

APPENDIX I: QUOTATIONS AND ADDITIONAL SPECIFICATIONS

In this appendix, quotations for a Vivitro system, PCI 30GA card, LabVIEW software and electrical-to-pressure regulator for the cylinder positioner are given. Full specifications of PCI30GA card and electrical-to-pressure regulators are also given as attachments.

QUOTATION FOR A VIVITRO SYSTEM: From an e-mail:

Dear Sir

Thank you for your enquiry. I believe our SUPERPUMP SYSTEM SPS3891 would meet the requirements you indicated. The motion of the pump can be programmed using our VIVIGEN WAVEFORM GENERATOR VG2001 in conjunction with a PC. The pressure conditions in the system attached to SPS3891 will depend on both the motion of the pump and the load on the pump. Our product VIA7991 (Viscoelastic Impedance Adapter) can be used to interface between pump and system and provide impedance matching.

Current prices are:

SPS3891: USA\$13,800

VG2001: USA\$2,400

VIA7991: USA\$2,100 and current delivery is 2 weeks.

Should you require further information, please do not hesitate to contact us.

Regards, David walker

QUOTATION FOR A PCI 30GA CARD

EAGLE Technology | Quote

Regards,

Henry Hugo

Eagle Technology

Tel : +27 21 4234943 Fax : +27 21 4244637

P O Box 4376, Cape Town 8000, South Africa

35 Hout Street, Cape Town 8001, South Africa

email : hugo@eagle.co.za

web : www.eagle.co.za

Copyright Notice:

The information contained in this E-mail message is confidential and may be legally privileged. It is intended solely for the addressee. Access to this E-mail message by anyone else is unauthorised. If you have received this E-mail message in error, please notify us immediately by telephone on +27 21 423 4943 or Email on eagle@eagle.co.za. Please also destroy and delete the message from your computer. If you are not the intended recipient, any disclosure, copying, distribution, modification in any form or reproduction or any action taken or omitted to be taken in reliance on it is strictly prohibited and may be unlawful.

-----Original Message-----

From: Henry Hugo [mailto:hugo@eagle.co.za]

Sent: 01 September 2003 02:46

To: myeoman@ebe.uct.ac.za

Subject: EAGLE Technology Quotation - EAGLE4764

Dear Mr I Pitiri

The pricing of the PCI-30GA is below.

Delivery would normally be ex stock.

A data sheet is attached.

You need to bear in mind that the output of the D/A converter will need to be put through an amplifier of some kind to drive the solenoid valve.

Regards,

Henry Hugo
hugo@eagle.co.za

31-35 Hout Street, Cape Town 8001, South Africa
PO Box 4376, Cape Town 8000, South Africa
Tel : +27 21 423 4943 Fax : +27 21 424 4637

email : hugo@eagle.co.za Web : www.eagle.co.za

General Details:

Company: UCT - Electro Mechanical Engineering
Vat No:

Attention : I Pitiri
Telephone: +27 21
Fax : +27 21
Delivery Address:

Quotation Reference Nr: EAGLE4764
Order Nr:
From : Henry Hugo (hugo@eagle.co.za)
Date : 2003-09-01

Item No.	Qty.	Product Code	Description	Unit Price,
				Ex VAT
				Extended
				Ex VAT

1.1	PCI-30GA	100kHz 16 Channel 12-bit A/D + four 12 Bit DACs	R
4,628.00			R 4,628.00

2.1	SCSI-C50MDB50F	50way SCSI-II Centronics (M) to DB50 (F) - 1m	R
175.00			R 175.00

3.1	ADPT-5050	DB50 (M) & IDC50 (M) to 51-way Screw Terminal	R
550.00			R 550.00

Sub Total: R 5,353.00

Vat: R 749.42

Total: R 6,102.42

Banking Details:
ACCOUNT NAME.....: EAGLE TECHNOLOGY
BANK.....: NEDBANK
BRANCH.....: BUSINESS SOUTHERN PENINSULA
BRANCH CODE.....: 123-209
ACCOUNT NO.....: 1031339183

CONDITIONS OF SALE

Payment Terms: 30 days

Warranty: All equipment is covered by a 1-year warranty, unless equipment has been subjected to misuse or tampered with.

Liability: Eagle Technology will not be held liable for any damage to body or material during installation of equipment. During the 1 Year Warrantee we will endeavour to do our best to support the system, however we will not be held liable for downtime on the system. We will also not be held liable for any misuse of equipment that may end in a court action.

Ownership: Goods remain the property of Eagle Technology until totally paid for.

Delivery : Ex stock.. Delivery quoted by supplier is in good faith and considered reliable for the day of quotation. The delivery date quoted is not guaranteed and will only be confirmed once our supplier has accepted our order and confirmed a supply date.

Validity : This quotation holds firm for 30 days from even date.

Approval of this document constitutes acceptance of this Quotation and Conditions of Sale

3x 24bit Counter Timers



Features

- 8/16 Differential or 16/32 Single-Ended A/D inputs
- 100kHz Sampling rate
- Analog input gains software selectable from: 1, 10, 100, 1000
- Unipolar or bipolar in signals for PCI-30GA
- Channel list FIFO(256 bytes) onboard for auto A/D channel scanning
- 24 Digital I/O lines in 3 ports (8255PPI)
- 3 User Counter/Timers to measure frequency/speed (Max Input: 10Mhz)
- Optional simultaneous sample hold models available
- Windows98/ME/2000/XP OS Support (NT on request)
- Linux OS Support
- WaveView for Windows Data Acquisition & Logging Software
- Labview, Testpoint and VEE Pro Drivers
- PCI-bus revision 2.2 compliant

Description

The PCI-30G series is a family of 12-bit data acquisition boards for PCI-based systems. The PCI-30G comes in a variety of versions, with or without D/As. The 'A' versions include four 12bit DACs with current sensing outputs.

The PCI-30G has uniquely flexible digital I/O capabilities consisting of 24 lines in three ports. Each port can be configured as inputs or outputs. It also supports hardware handshaking, strobe I/O or bi-directional protocols.

The PCI-30G also includes six 16-bit user counter/timers. Two are used by the PCI-30G circuitry, four being uncommitted and available to the user to measure frequency and count events or speed. A/D and D/A signals are available on 50-way SCSI-Centronics connector. The DIO and counters come out on the IDC40 header.

SCSI-II-50F Cent

CHAN31	50	25	CHAN30
CHAN29	49	24	CHAN28
CHAN27	48	23	CHAN26
CHAN25	47	22	CHAN24
CHAN23	46	21	CHAN22
CHAN21	45	20	CHAN20
CHAN19	44	19	CHAN18
CHAN17	43	18	CHAN16
CHAN15	42	17	CHAN14
CHAN13	41	16	CHAN12
CHAN11	40	15	CHAN10
CHAN9	39	14	CHAN8
CHAN7	38	13	CHAN6
CHAN5	37	12	CHAN4
CHAN3	36	11	CHAN2
CHAN1	35	10	CHAN0
AGND	34	9	AGND
AGND	33	8	AGND
SENSE0	32	7	DAC0
SENSE1	31	6	DAC1
SENSE2	30	5	DAC2
SENSE3	29	4	DAC3
EXT CLK	28	3	DGND
+12V	27	2	EXT TRIG
+12V	26	1	+5V

IDC-40M

PA0	1	2	PA1
PA2	3	4	PA3
PA4	5	6	PA5
PA6	7	8	PA7
PB0	9	10	PB1
PB2	11	12	PB3
PB4	13	14	PB5
PB6	15	16	PB7
PC0	17	18	PC1
PC2	19	20	PC3
PC4	21	22	PC5
PC6	23	24	PC7
DGND	25	26	CLK2
CNT0	27	28	OUT2
CGTE0	29	30	CGTE1
CGTE1	31	32	CNT1
CNT2	33	34	CNT1
CNT2	35	36	CGTE2
+5V	37	38	DGND
DGND	39	40	DGND

DB-37M (Internal Cable)

PA1	20	1	PA0
PA3	21	2	PA2
PA5	22	3	PA4
PA7	23	4	PA6
PB1	24	5	PB0
PB3	25	6	PB2
PB5	26	7	PB4
PB7	27	8	PB6
PC1	28	9	PC0
PC3	29	10	PC2
PC5	30	11	PC4
PC7	31	12	PC6
CLK2	32	13	DGND
OUT2	33	14	CNT0
CGTE0	34	15	CGTE0
CNT1	35	16	CGTE1
CNT1	36	17	CNT2
CGTE2	37	18	CNT2
+5V		19	+5V



Supplied with Internal Cable for Digital I/O (IDC40 to DB37)

Specifications

ANALOG INPUTS (A/D)	
Input Characteristics	
Input channels:	16/32 Single-Ended or 8/16 Differential
Overvoltage Protection:	±20V(powered on); ±35V(powered off)
Resolution:	12-bit (1 in 4096)
Input Ranges:	±5V, ±10V, 0 to 10V
Input Coupling:	DC
A/D Transfer Characteristics	
System Accuracy:	±1 LSB depending on environment
A/D Linearity	
Differential:	±¾ LSB max
Integral:	±0.05%FS
SNR:	84dB typ
Full BW:	1MHz
Total Harmonic Distortion:	-98dB
System Accuracy:	±1 LSB depending on environment
Acquisition Rate:	100kHz max
A/D FIFO buffer:	2/4/8/16/32K (optional)
Acquisition Modes:	Polled I/O, Interrupts, REP INSW
Amplifier Characteristics	
Input Impedance:	10G Ohm/20pF (On chan) 10G Ohm/100pF (Off chan)
Offset voltage: Adjustable to zero	
Input Gains	
Ranges:	1, 10, 100, 1000
Error:	Adjustable to 0
Non-linearity:	0.002% (typ), 0.015% (max) [G<1000] 0.02% (typ), 0.06% (max) [G=1000]
Gain Accuracy:	0.25% max, 0.05% typ
CMRR:	100dB typ, 80dB max for G=1
Monotonicity:	0 to 70°C
Temperature drift:	6ppm/°C (Full Scale) 1ppm/°C (Bipolar zero) ±30ppm/°C (Gain typ)
Dynamic Characteristics	
Bandwidth(small signal):	1.0MHz (G<1000) 250kHz (G=1000)
Full Power Bandwidth:	1MHz for G<1000, 100kHz for G=1000
Crosstalk:	-85dB, DC to 100kHz
System Noise:	<0.5LSB (G=1) ±2LSB (G=10) ±4LSB (G=100)
ANALOG OUTPUTS (D/A)	
No of Channels:	4x 12-bit
Accuracy:	±1 LSB
DNL:	½ LSB max
Output Ranges:	±5V, ±10V, 0 to 10V, 0 to 13V
Thruput Rate:	100kHz (depending on computer)
Offset Error:	Unipolar: ¼ LSB typ, 1 LSB (max) Bipolar: ½ LSB typ, 2 LSB (max)
DIGITAL I/O (DIO)	
No of TTL I/O lines:	24 in 3 ports (8255 PPI)
Digital Logic Levels:	High: 2.0V (min), 5.0V (max) Low: 0.0V (min), 0.8V (max)
Current Output:	±3mA (source/sink)
Interrupt support:	Yes (Mode 0, Mode 1, Mode 2)
External Interface	
Connector Types:	SCSI 50-way Right Angle Female Centronics-Type Connector IDC40 Header (for Digital I/O)
Counter Timers	
Resolution:	16-bit
Clock Frequency:	2 or 8 Mhz (for A/D)
A/D Frequency:	DC to 330kHz
No of counters:	6 (3 used for A/D conversion)
User Pins:	4 Input CLKs, 3 Gates & 4 Outputs
Compatibility:	TTL
PCI Interface	
Base Address:	Auto selected
Number of registers:	64x 16-bit registers
Interrupts:	Auto selected
Environmental / Physical	
Relative Humidity:	0% to 90% (non-condensing)
Operating Temp:	0°C to 70°C
Board Dimensions:	193mm x 111mm
Power Requirements	
+5V	1.2A typ
+12V	150mA (S models only)
-12V	150mA (S models only)

Ordering Information

Supplied with EDR Enhanced Software and Internal Cable for Digital I/O (IDC40 to DB37)

All boards have A/D Inputs, 24 Digital I/O lines + four 16-bit counters

PCI-30G	100kHz 16 Channel A/D
PCI-30GA	100kHz 16 Channel A/D + four 12-bit DACs
PCI-30G-32	100kHz 32 Channel A/D
PCI-30GA-32	100kHz 32 Channel A/D + four 12-bit DACs

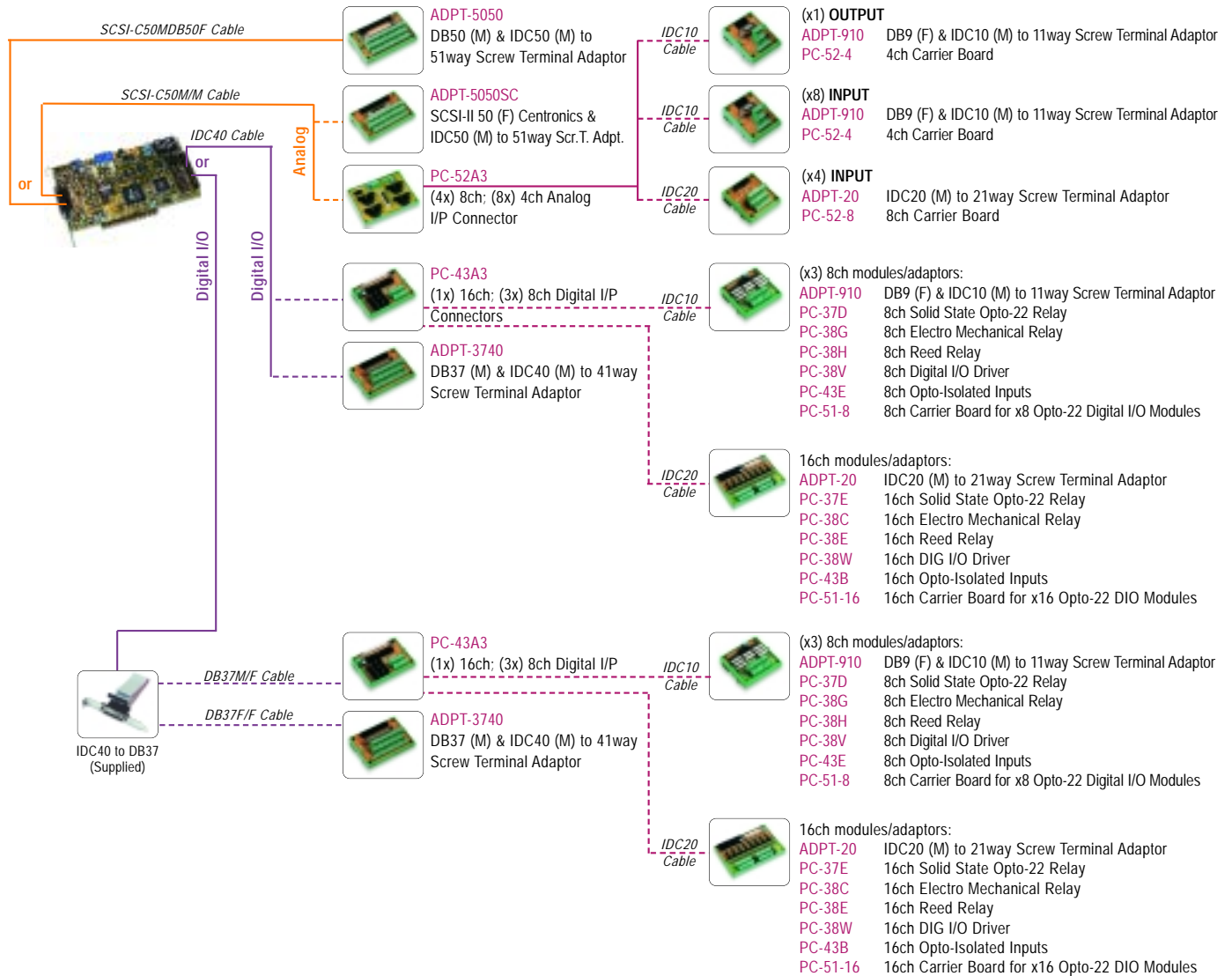
Optional Accessories

OPTION 1:	
ADPT-5050	DB50 (M) & IDC50 (M) to 51way Screw Terminal Adaptor
SCSI-C50MDB50F	50 SCSI-II Centronics (M) to DB50 (F) Screened Cable

OPTION 2:	
ADPT-5050SC	SCSI-II 50 (F) Cent & IDC50 (M) to 51way Screw Term. Adaptor
SCSI-C50M/M	50 SCSI-II Centronics (M) to (M) Screened Cable

OPTION 3:	
ADPT-3740	DB37 (M) & IDC40 (M) to 41way Screw Terminal Adaptor
DB37F/F	DB37 (F) to DB37 (F) Screened Cable

Optional Accessory Diagram



FREE WaveView for Windows Software



WaveView for Windows is a new Microsoft® Windows™ based data acquisition package supporting our PCI range of personal computer plug-in cards. The software is extensively configurable and easy to use. The WaveView for Windows software package is used for collecting and analyzing data. Two modes of operation are supported, **scope mode and chart recording**. WaveView can also be used as a **waveform generation tool**, or a **digital power supply controller**. The software is extensively configurable, easy to use and quick to learn.



QUOTE

National Instruments ZA
 Block A15 Greenoaks Office Park
 Cnr Bekker Rd & Gregory Ave.
 VORNA VALLEY 1686

Tel: 011 805-8197
 Fax: 011 805-8194

ATTN: eMail
 Itumeleng Pitiri <PTRITU001@mail.uct.ac.za
 Dept. Mechanical Engineering
 UCT
 Tel: 072 361 7036
 Fax: 021 650 3240

TOLLFREE: 0800 203 199

T/A: **NI Solutions (Pty) Ltd.**

QUOTE	zaq030550	08-Sep-03	Payment Terms:	PREPAYMENT
Shipping Terms: Will ship on presentation of Bank Deposit Slip - unless Net 30 Credit Application Completed				
Special Terms and Conditions:				

We are pleased to submit this quotation for your consideration. Please specify the quotation number when ordering.

Line No.	Part No.	Description	QTY	Unit Price	Academic Discount	Extended Amount
01	776670-03	LabVIEW Full Development System for Windows 2000/NT/Me/9x CD contains GPIB,DAQ,RS-232, and Adv. Analysis VI Libraries and CIN Development Toolkit.	1	R 20,928.00	75%	R 5,232.00

- * Prices are exclusive of VAT. * Validity: 30 Days.
- * The Price of the above items includes all applicable duties at present. Any changes in the VAT, Customs Surcharge, or Ad Valorem, which may be imposed by the authorities between the date of placing your order and actual delivery, will be for your account.
- * Banking Details: Standard Bank, Randburg, Branch Code 018005, Account No. 021664749. Account name: NI Solutions (Pty) Ltd.
- * Terms: On Payment - Please fax bank statement to National Instruments South Africa.
- * Delivery: 2 weeks. * Prices are Fixed and Firm.

Yours sincerely, Michael Hutton for National Instruments Southern Africa.

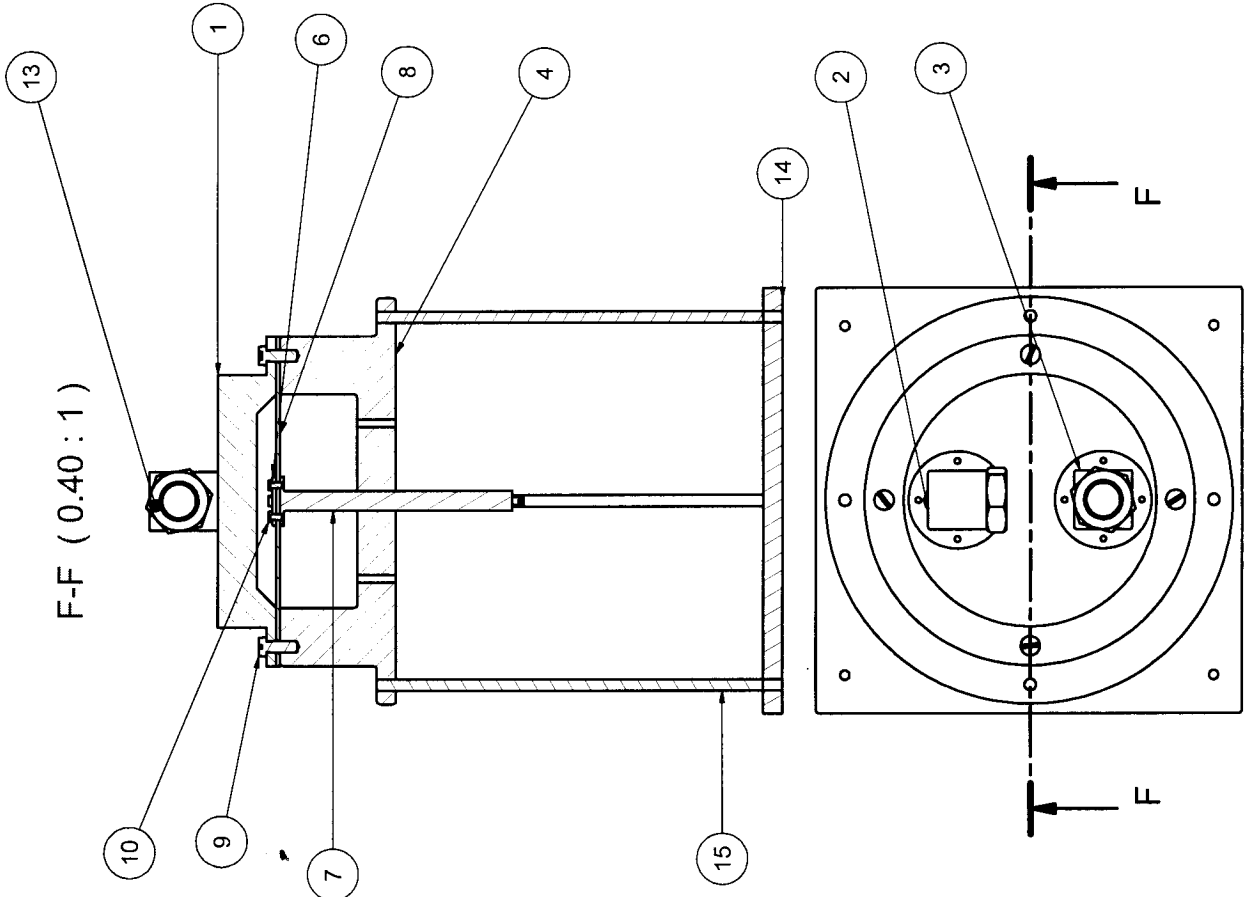
Terms and Conditions of Sale

All sales are subject to acceptance of National Instruments terms and conditions of sale. By placing an order, you agree that National Instruments shall not be bound by any conflicting additional terms and conditions. You may obtain a complete copy of National Instruments terms and conditions of sale by calling our local office **TOLLFREE 0800 203 199**

FOR QUESTIONS CONCERNING THIS QUOTE PLEASE CONTACT NATIONAL INSTRUMENTS AT +27 11 805 8197
 National Instruments trading as NI Solutions (Pty) Ltd (PO Bcx 10931, Vorna Valley, 1686 Midrand) is wholly owned by NI Corp. USA

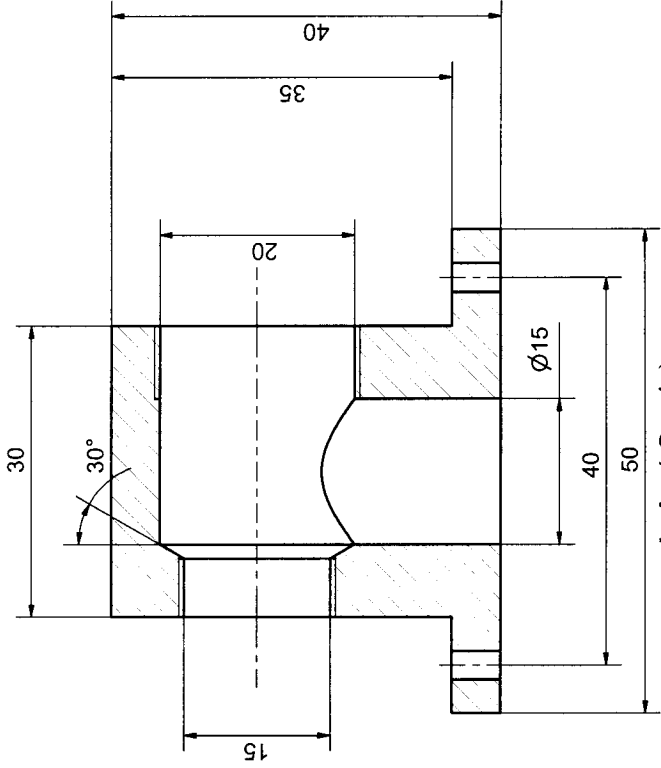
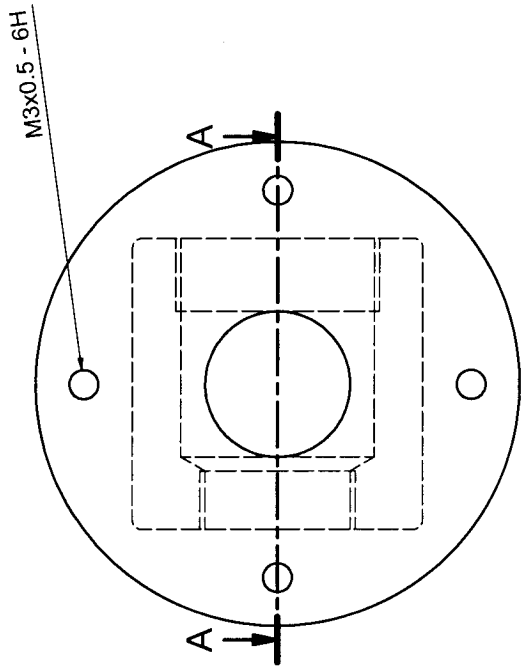
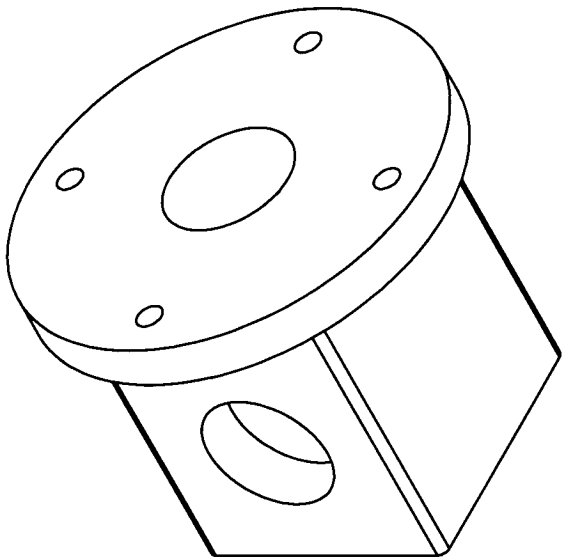
APPENDIX J: ASSEMBLY AND PARTS DRAWING OF
PUMP

N.B: An adjustable bolt is not a standard part. It is to be made from a nut and a cylindrical shell with outside threads. (They are to be welded together)



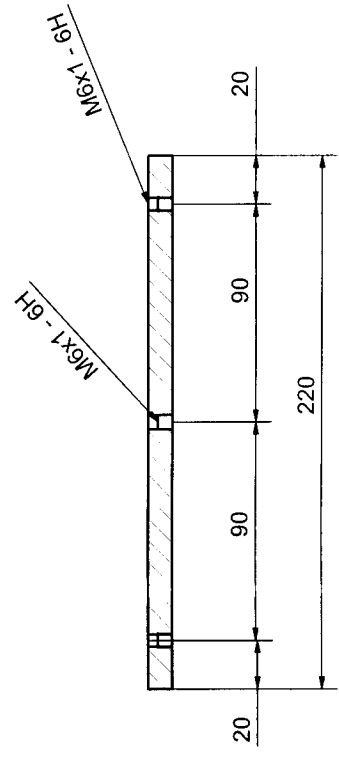
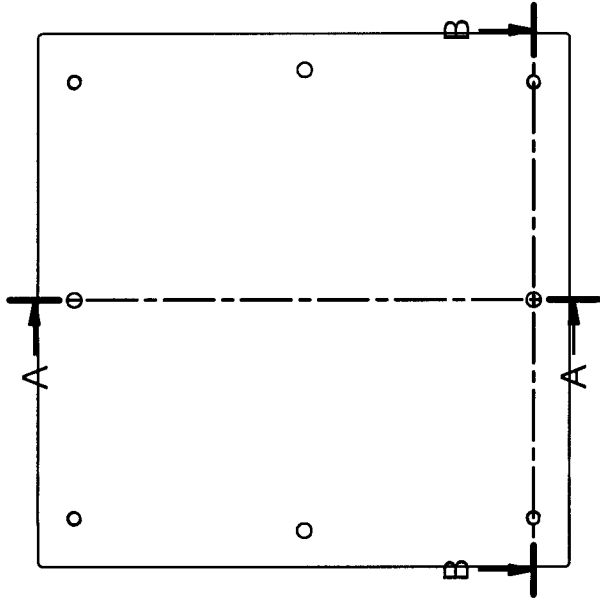
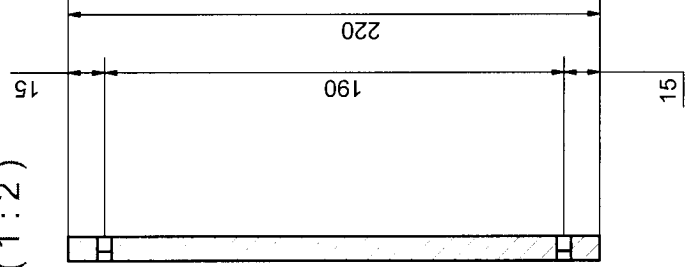
Parts List			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Diaphragm_top	Teflon
2	1	Inlet_housing	Teflon
3	1	Outlet_housing	Teflon
4	1	Pump_base1	Teflon
6	1	Piston_cap	Stainless steel
7	1	Piston_b_cap	Stainless steel
8	1	Elastic membrane	Silicon (medical grade)
9	4	ISO 1207 - M 6 x 16	Slotted Cheese Head Screw
10	4	ISO 1207 - M 3 x 5	Slotted Cheese Head Screw
11	2	Ball valve	Stainless steel
12	2	Adjustable bolt	Stainless steel
13	2	ISO 4035 - M20	Hex Nut
14	1	Mounting_base	Mild steel
15	4	Mounting_rod	Mild steel

Designed by student
 Checked by _____ Approved by _____ date 10/23/2003
 Date 10/23/2003



Designed by student	Checked by	Approved by - date	Date 10/31/2003
inlet_housing(no spring)			Sheet 1 / 1

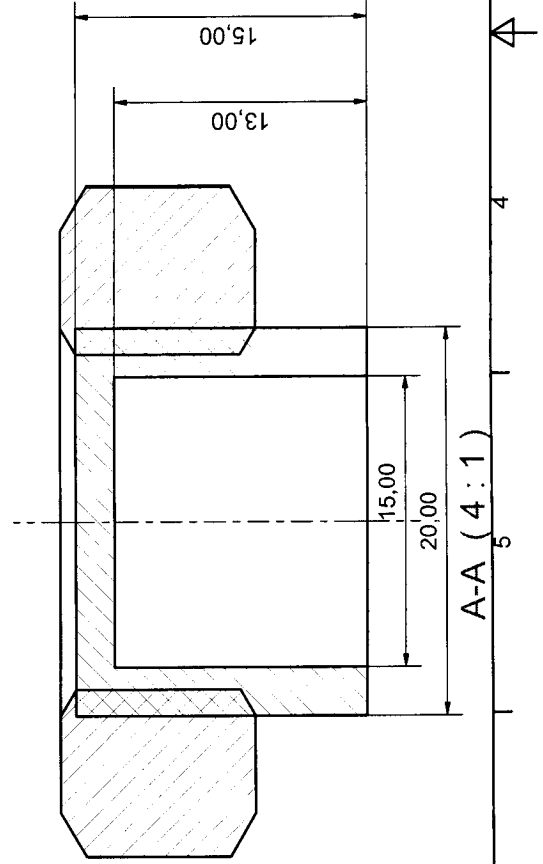
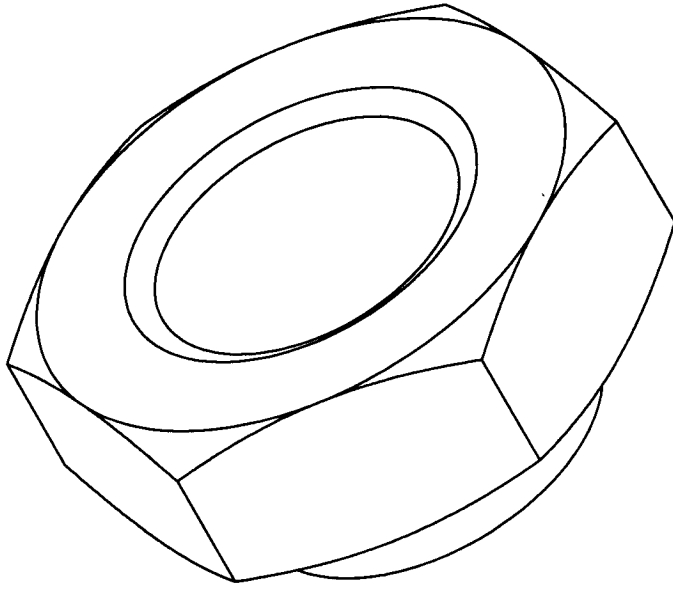
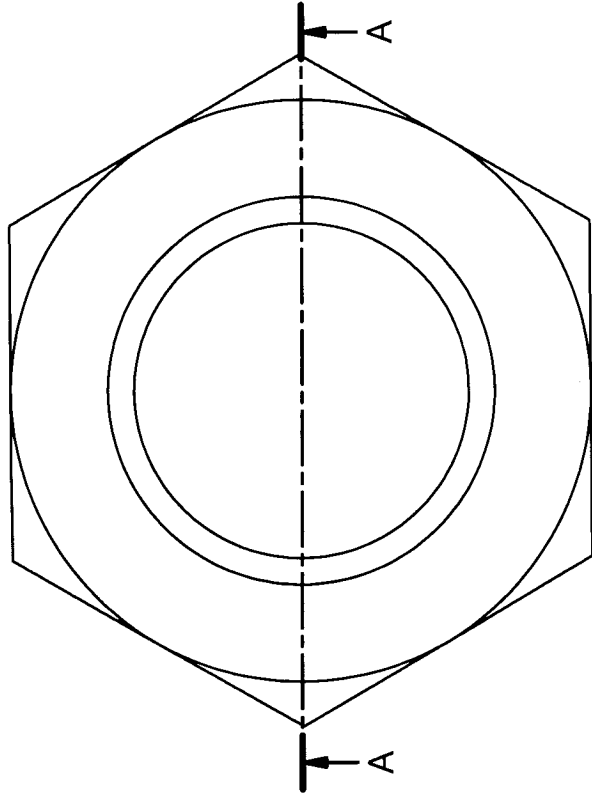
A-A (1:2)



B-B (1:2)

Designed by student	Checked by	Approved by - date	Date 10/31/2003
Mounting_base			Edition 1 / 1

N.B: This is not a standard part. A nut and a threaded cylinder must be welded together



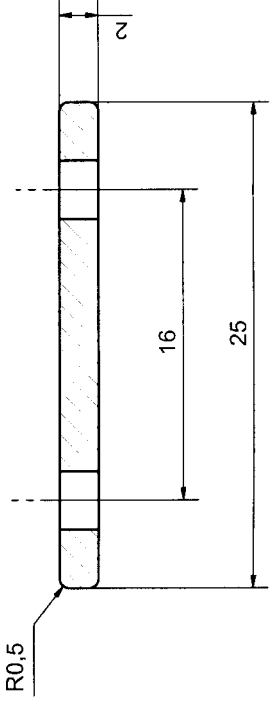
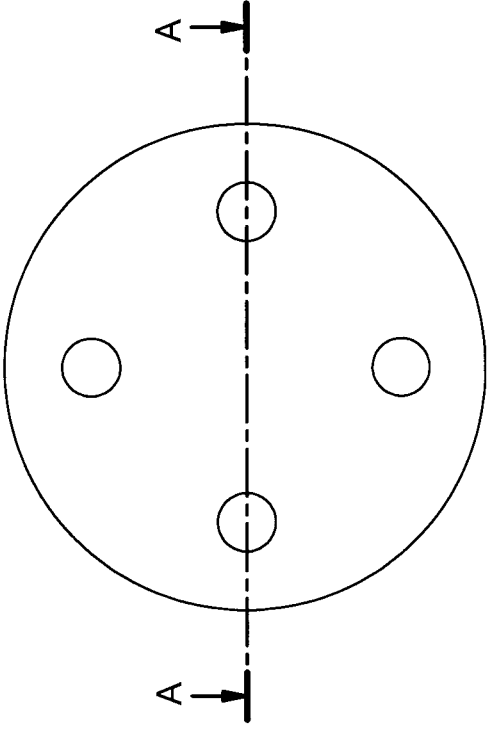
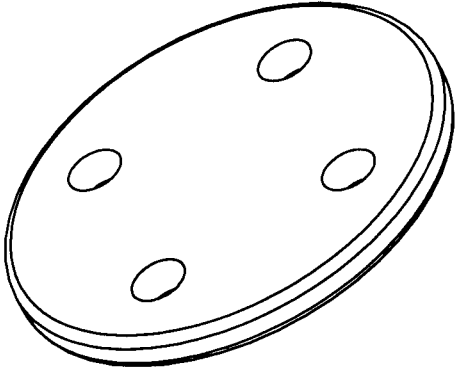
Parts List			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Adjustable bolt	Stainless steel
2	1	ISO 4035 - M20	Hex Nut
Designed by		Checked by	Date
PITIRI, I		Approved by - date	10/25/2003

Adjustable bolt

Edition

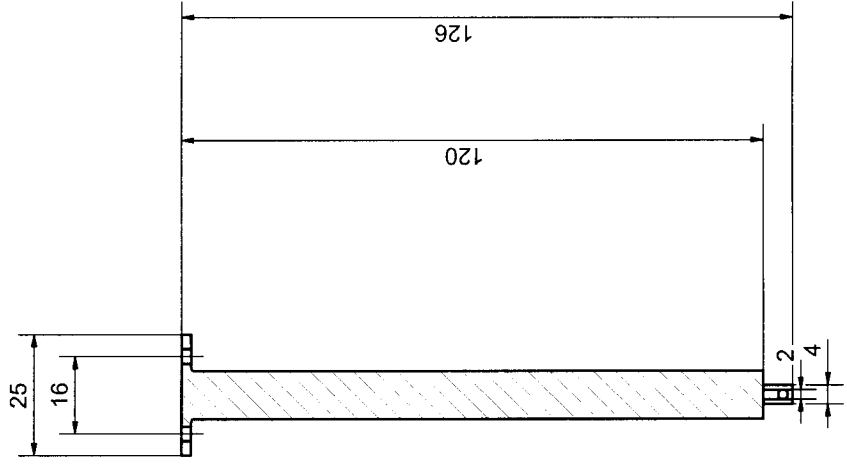
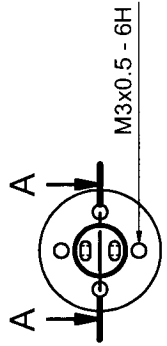
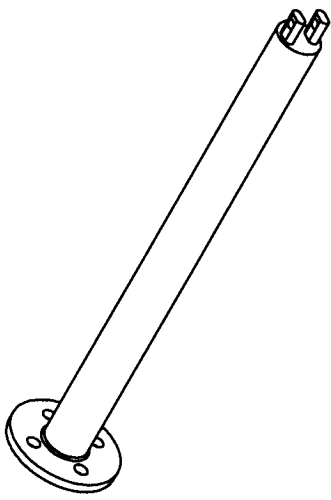
Sheet

1 / 1



A-A (4:1)

Parts List			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Piston_cap	Teflon
Designed by student		Checked by	Date 10/24/2003
		Piston_cap	
		Edition 1 / 1	



A-A (1:1)

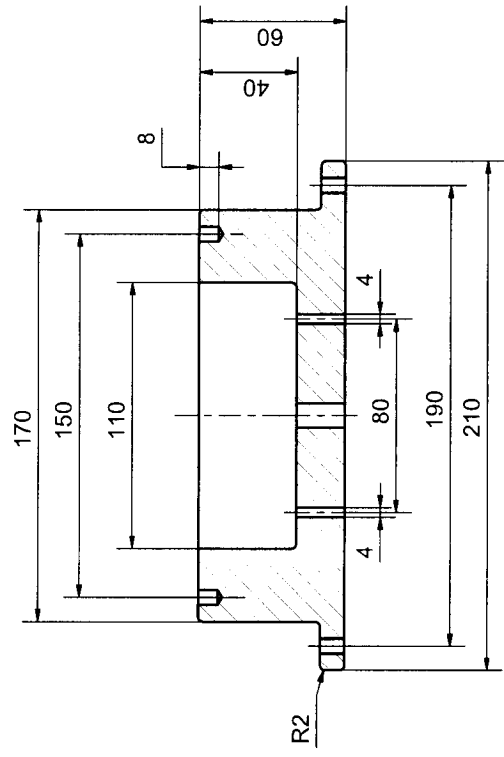
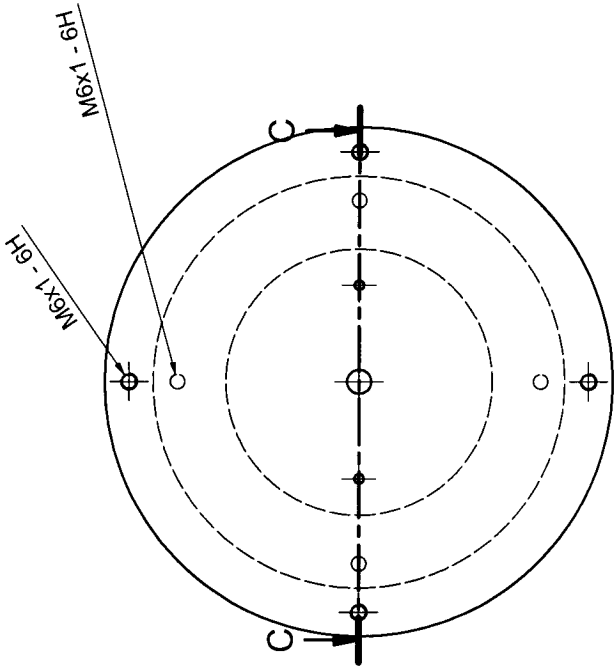
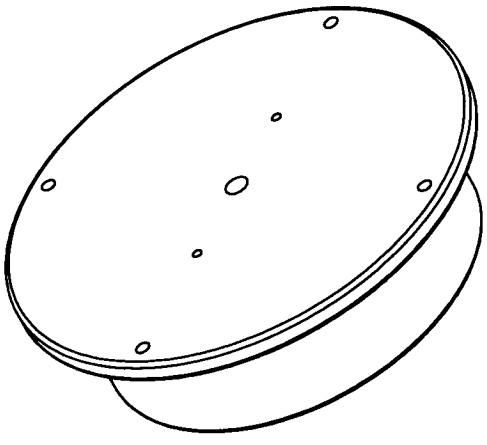
Parts List			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Piston_b_cap	Stainless steel
Designed by		Checked by	Date
student			10/24/2003

Piston_b_cap

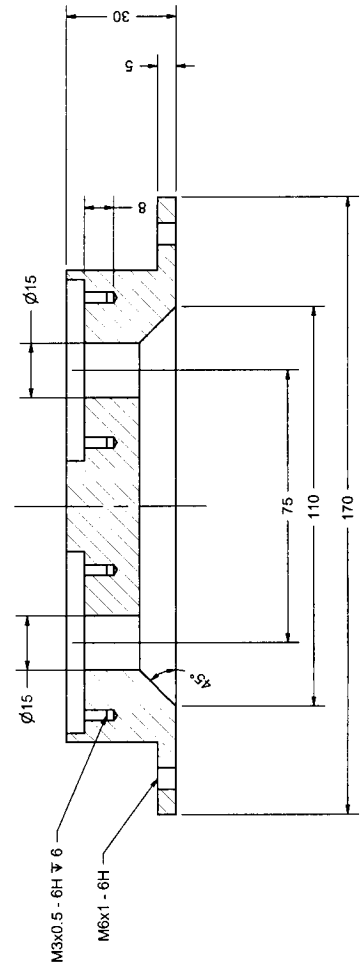
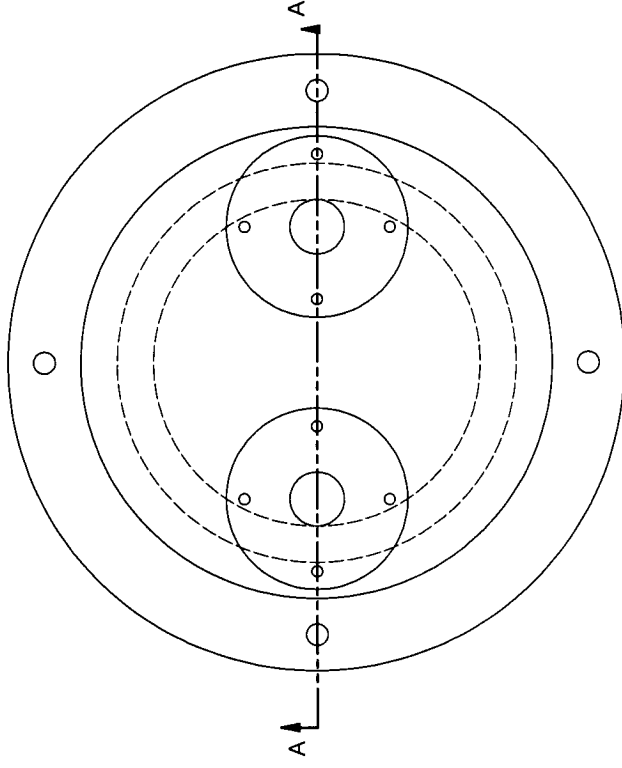
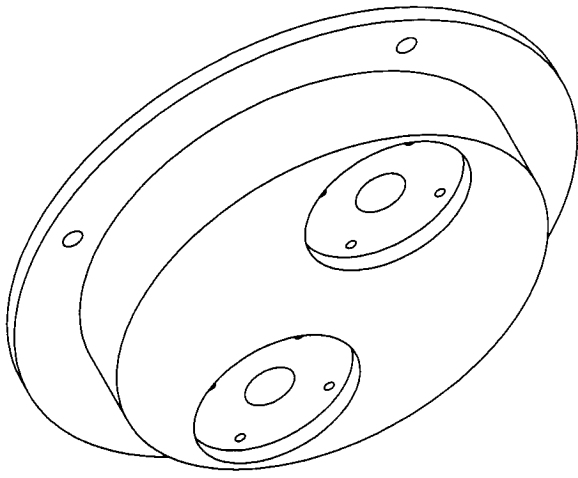
Edition

Sheet

1 / 1



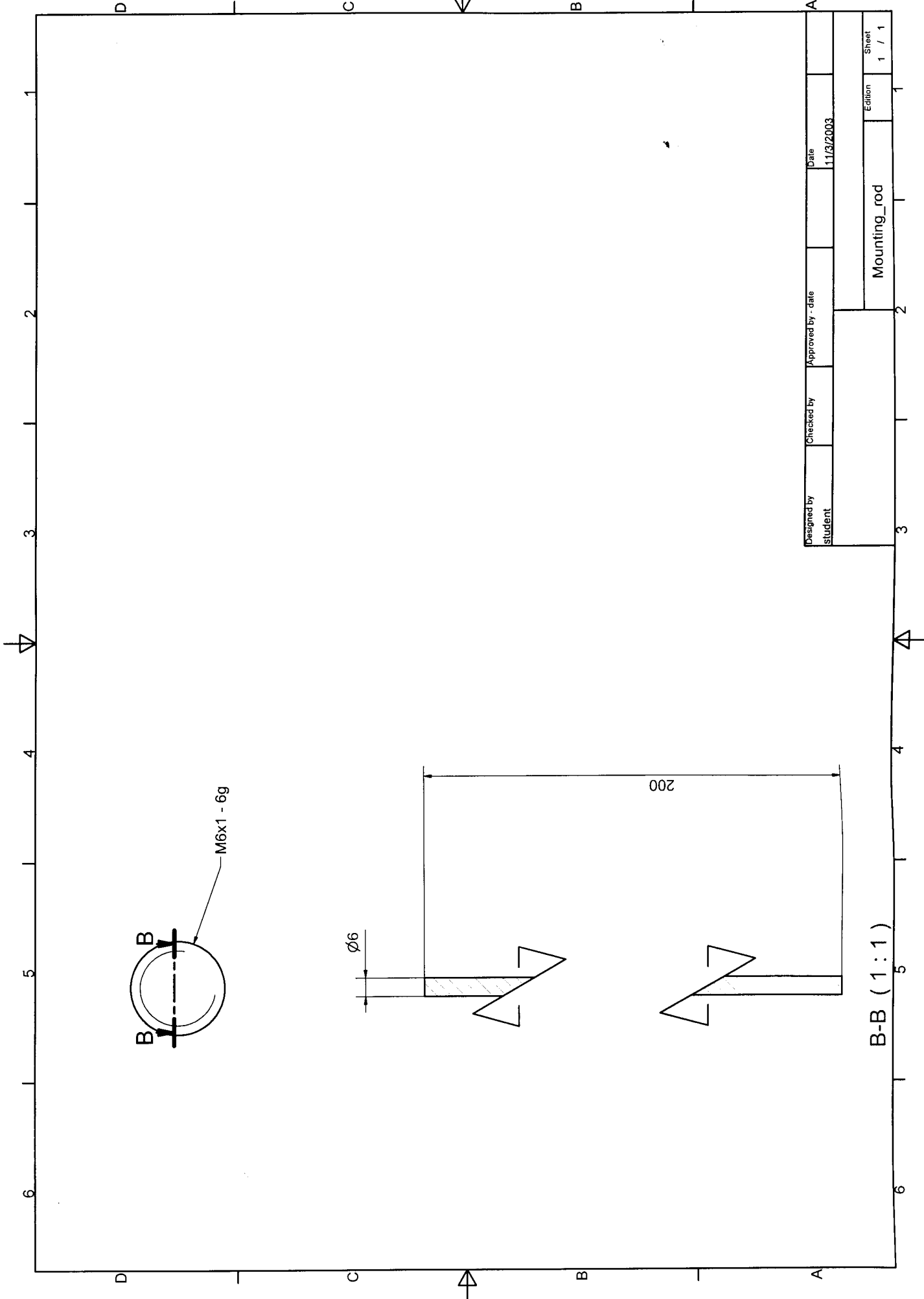
Designed by student	Checked by	Approved by - date	Date 10/14/2003
Pump_base1_s			Edition 1 / 1
Sheet 1 / 1			



SECTION A-A
SCALE 1 : 1

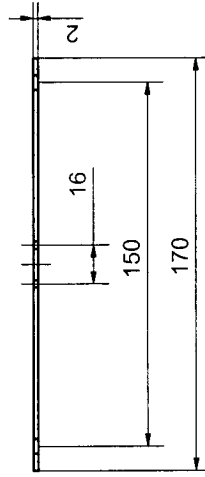
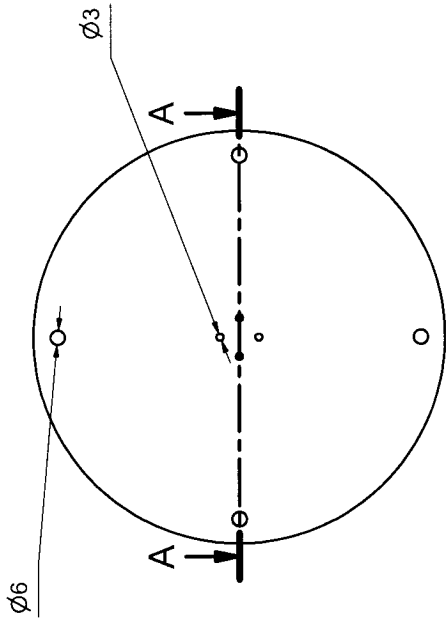
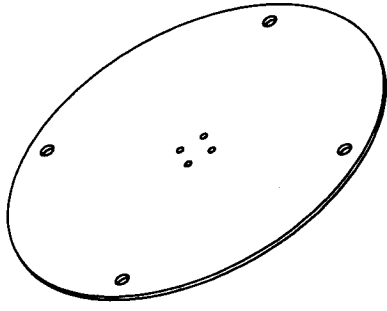
Parts List		
ITEM	QTY	DESCRIPTION
1	1	Diaphragm_top
Teflon		

DRAWN	21/10/2003	TITLE	
PLIN			
CHECKED			
QA			
MFG			
APPROVED			
SIZE	C	DWS NO	Diaphragm_top
SCALE		REV	
		SHEET	1 OF 1



Designed by student	Checked by	Approved by - date	Date 11/3/2003
Mounting_rod			Sheet 1 / 1

B-B (1:1)



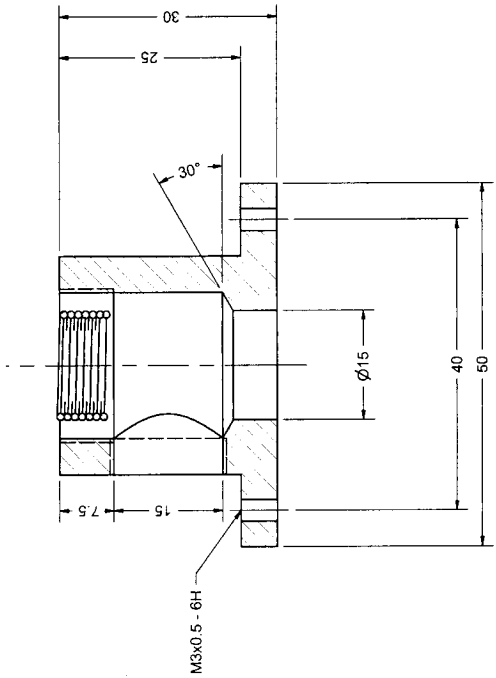
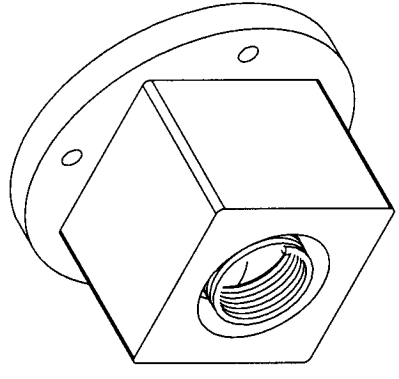
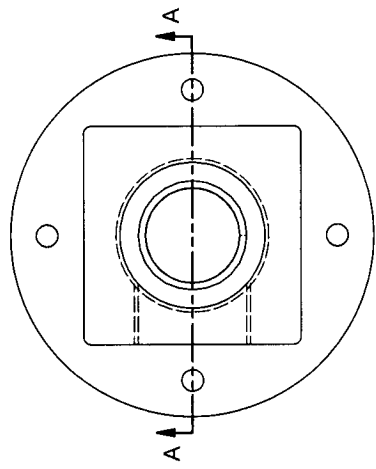
A-A (1 : 2)

Parts List			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Elastic membrane	Silicon (medical grade)
Designed by Student		Checked by	Date
		Approved by	10/23/2003

Elastic membrane

Sheet 1 / 1

The drawing shows how the spring shall be located in the housing



SECTION A-A
SCALE 2 : 1

DRAWN	21/10/2003	TITLE	
PHIS		SIZE	C
CHECKED		DWGS NO	Outlet housing
QA		REV	1
MFG		SCALE	1 OF 1
APPROVED			