DEVELOPMENT OF A SOFTWARE FOR DRAFTING AND SIMULATION OF OPEN LOOP HYDRAULIC POWER CIRCUITS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET AKİF ERŞAHİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

IN

THE DEPARTMENT OF MECHANICAL ENGINEERING

SEPTEMBER 1996

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Tayfur ÖZTÜRK
for Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ediz PAYKOÇ
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Y. Samim ÜNLÜSOY
Supervisor

Examining Committee Members:

Prof. Dr. Ömer ANLAĞAN

Prof. Dr. A. Demir BAYKA

Prof. Dr. O. Cahit ERALP

Prof. Dr. Y. Samim ÜNLÜSOY

Prof. Dr. Can ÇOĞUN

ABSTRACT

DEVELOPMENT OF A SOFTWARE FOR DRAFTING AND SIMULATION OF OPEN LOOP HYDRAULIC POWER CIRCUITS

Erşahin, Mehmet Akif

M.S., Department of Mechanical Engineering

Supervisor: Prof. Dr. Y. Samim Ünlüsoy

September 1996, 91 pages.

A software, DRAFTSIMA, is developed containing libraries of standart fluid power symbols for interactively constructing schematics of circuits and simulating operation of the circuits constructed.

Standart graphic symbols are taken from ISO 1219-1 [1] (TS 1306 [2]), and a symbol library has been prepared and classified according to the functions of the components. Any desired circuit diagram can be constructed by selecting the related components from the menus of the program visually, and the schematics can be plotted or printed on paper with additional comment texts. Once an open loop hydraulic fluid power circuit is drawn, software simulates operation by animating functions of components: valves shift from one state to another, cylinders extend, hold or retract, pressure control valves opened according to their set values, hydraulic motors rotate according to valve position, fluid lines change color to indicate fluids pressure state. Position of the valves can be controlled by user or by pilot pressure lines controlling the pressure value at certain points of the system. Pressure value at any point of the circuit or velocity of actuators can be obtained using proper sensor elements. Program is limited to steady state simulation of open loop circuits.

Keywords: Open-Loop Hydraulic Circuit, Drafting, Simulation, Hydraulic Power Symbols, Steady-State Simulation.

ÖZ

AÇIK ÇEVRİM HİDROLİK GÜÇ DEVRELERİNİN ÇİZİMİ VE SİMULASYONU İÇİN BİR YAZILIM GELİŞTİRİLMESİ

Erşahin, Mehmet Akif
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Y. Samim Ünlüsoy

Eylül 1996, 91 sayfa

Hidrolik güç devre şemalarının çizimi ve oluşturulan açık çevrimli devrelerin çalışmasının simulasyonu için standart devre elemanlarına ait sembol kütüphanesi içeren DRAFTSIMA adında bir yazılım geliştirilmiştir.

Devre elemanlarının standart grafik sembolleri ISO 1219-1 (TS 1306) dan alınmış ve elemanlar işlevlerine göre sınıflandırılarak ilgili başlıklar altında sembol kütüphanesi oluşturulmuştur. İstenen akışkan gücü devresi, yazılımın menülerinden görsel olarak elemanlar seçilerek oluşturulabilmekte, oluşturulan şemalar açıklayıcı notlarda eklenerek kağıda aktarılabilmektedir. Açık çevrimli bir akışkan gücü devresi oluşturulduğunda, yazılım devrenin çalışmasını elemanların fonksiyonlarını yerine getirerek simüle edebilmektedir.Bu amaçla akış kontrol valfleri durum degiştirerek akış yolunu belirlemekte, pistonlar ileri-geri hareket etmekte, motorlar dönmekte, basınç kontrol valfleri verilen değere göre akış yönünü belirlemekte, boruların rengi taşıdıkları akışkanın basıncına göre degişmektedir. Akış kontrol valflerinin durumları

simulasyon sürecinde herhangi bir anda kullanıcı tarafından değiştirilebilmekte yada basınç sinyali hatları sayesinde devrenin istenen noktasındaki basınç durumuna göre yazılım tarafından belirlenmektedir. Sonuç olarak istenen açık çevrimli sistemin davranışının gözlenebilmesi mümkün olmaktadır. Devrenin herhangi bir noktasındaki basınç değerine yada pistonların hızlarına kullanıcı tarafından ilgili ölçüm elemanları kullanılarak ulaşılabilir. Yazılım açık çevrimli sistemlerin sürekli rejim simulasyonu ile limitlidir.

Anahtar Kelimeler : Açık Çevrimli Hidrolik Güç Devresi, Çizim, Simulasyon, Grafik Sembol, Sürekli Rejim.

ACKNOWLEDGMENTS

Bana yaşamı sevdiren kişilere teşekkür etme firsatını yakaladığım bu satırları, duygularımı yaşadığım gibi yansıtabilmek için ana dilimde yazmak isterdim.

Herkesden önce, lisans eğitimimde aldığım dersleri sayesinde tanıdığım, mastır yapma kararını kendisiyle çalışabilmek için aldığım ve birlikte çalışıyor olmaktan gurur duyduğum, son üç yıldır hayatımın her alanında beni destekleyen, her konuşmasından birşeyler öğrendiğim, akademik çalışmayı ve mühendisliği bana sevdiren, bütün yaşamım boyunca kendime örnek olarak aldığım tek insan olan tez danışmanım Prof. Dr. Y.Samim Ünlüsoy'a yürekten teşekkür ederim.

Yine son üç yılda kazandığım en değerli varlık olarak gördüğüm, çalışmalarımda bana sınırsızca yardım eden, zor dönemlerimde hep yanımda bulduğum ortağım ve "yakın dostum", fedakar insan Canan Yılmazel' e arkadaşlığıma katlandığı için teşekkür ederim.

I would like to express my thanks to Prof. Dr. O. Cahit Eralp for his support to my studies and daily life with his never ending smiles on his face.

Special thanks goes to Ali Özen Akyürek and Hüma Akyürek for their friendship and special support on computer hardware.

I am also grateful to my friends Çınar Gerçek, Okan Alkan, Dursun Öner, Ömür Menteş, Besim Baranoğlu. Working with them was so exciting and warm.

Finally, the greatest thanks go to my family, my brother Bülend, and my parents Gülten and Ekrem Erşahin.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The realization of a hydraulic power system is normally carried out through a number of stages.

i- Definition of the function to be performed by the system : To perform the expected operation from the hydraulic power system, the function expected from the operation of the system must be fully detailed with all the relevant specifications and the data.

ii- Drafting of the circuit: Hydraulic circuits are developed through the use of graphical symbols for all components. The symbol is intended to illustrate the type or function of the component, connections and the flow paths. The designer should know the symbol, and detailed operation characteristics of each element. Components are connected to each other by lines which represent conductors. Sometimes the circuit is re-drafted many times to obtain a satisfactory solution.

iii- Simulation of the operation: For different combinations of valve and load positions behavior of the system is analyzed. This is done by following the fluid path in the circuit schematic drafted. If unexpected situations are faced, the system is modified and schematic is redrafted until a satisfactory circuit is obtained.

iv- Selection of the components: According to the operation conditions specifications of each component (capacity of pumps, diameter and strokes of cylinders, set values of pressure valves, etc..) are determined. Then from manufacturers catalogs commercially available components are selected

v- Installation and commissioning : The system is constructed with the guidance of the schematic using the actual components. It is operated to prove satisfactory performance.

There exists a number of professional software related to fluid power applications [3,4,5]. However, each of these software are ment for either general use in the field or for specific applications. The general application have usually a number of limitations particularly in case of simulation. A typical example is HydrauSim [3]. This program is advertised for use in all fluid power systems, not only hydraulic or pneumatic systems. It also has the facility for the electrical controls. However, a close examination shows that the program is incapable of simulating accumulator circuits, circuits containing more than one actuator on the same branch, sequencing circuits, etc.. Thus, when one requires a tool for use, say, in education the requirements can not be satisfied by the available software in the market. This has lead to the development of a new software, DraftSima, for the drafting and simulation of hydraulic circuits.

The subject of the thesis is the preparation of a software to realize the second and the third stages of the realization process for a hydraulic power system incorporating the use of computer making it possible for the user to quickly and accurately draft and simulate the required circuit.

DraftSima, the developed software, contains a library of standard hydraulic power circuit elements. Symbols are classified under the headings according to their main functions (pumps, actuators, directional control valves, pressure control valves, flow control valves, accessories, conductors). Any required element symbol can be selected from the tables provided visually. Orientation of any component can be changed; a piston can be placed horizontally or vertically, from left to right or right to left, upside down or reverse. Different control types for directional control valves can be selected; lever, pilot operated, spring centered, solenoid, etc. so user can construct his/her own valve type among many alternatives. Pressure lines or pilot lines can be drawn simply by snapping grids, so it becomes easy to connect output of a valve to a piston or any other element. Components can be placed at any location on the drawing area of the screen by dragging with the mouse pointer. Position and orientation of the components can easily be changed at any time or they can be removed from the system.

For circuits containing many elements, with the zoom facility of the program, any portion of the schematic can be seen in detail or the parts going out the screen can be managed to fit the screen and be printed.

Descriptive texts can be added to the schematics and they can be saved for resuming any future study with the prepared circuit. DraftSima uses its own file format for keeping the drawings on disk files.

Schematics constructed by the program can be printed or plotted on paper. A title block can be attached on the printed form including job title, page number, date and time. With the print preview utility, before printing the schematic, final form of the printing on paper can be controlled. Standard paper sizes (A4, A3), and orientation (landscape or portrait) can be selected within the program.

Software is limited to the simulation of the steady state operation of open loop circuits. The operation of the system can be observed under different possible operational stages. Directional control valves shift position with user inputs. Then path of flow and behavior of elements within the system can be observed. Cylinders extend, hold or retract in compliance with the user supplied values (applied load, piston and rod diameters), relief valves open when the system pressure exceeds the set value, pilot lines send signals to directional control valves, pressure or flow control valves by checking pressure values at the points they attached making the simulation of automatically operating systems possible.

Software is basically ment to be used as an educational package. Open loop circuits given in text books or course notes of "Fluid Power Control " courses can be drafted and simulated. Number of components used in the system has no limitation for drafting and simulation, but computer's system configuration can limit the size of the circuit from the memory point of view.

DraftSima runs under Microsoft Windows 3.1 or later. It is designed to run standalone, no other CAD package is required. Visual Basic 3.0 Professional Edition is used as the programming language. Program uses all the advantages of operating system. Figures drafted on screen can be taken to any word processor or image editing software by using Windows clipboard. All hydraulic system schematics included in this thesis are drafted by DraftSima and taken in to document by standard copy-paste procedures of Microsoft Windows operating environment.

CHAPTER 2

HYDRAULIC POWER BASICS

In this chapter basic definitions and information hydraulic power systems are given. Types of components and their basic operation, main hydraulic power circuit types, special circuits and their working principles are given with explanatory figures. Usage and capabilities of DraftSima will be explained in the following chapter based on these basic concepts [6,7,8,9].

2.1 Fluid Power Systems

There are three basic systems for power transmission: fluid power systems, mechanical systems and electrical systems. Fluid power is the technology that deals with the generation, control, and transmission of power by means of pressurized fluids. Fluid power systems is used at most of the machines where pushing, pulling, holding a load, or rotating a shaft is required.

Fluid Power is the general term used to define such systems. The systems which use liquids (usually petroleum or synthetic oil) as the pressurized fluid are called ***Hydraulic Power Systems***. On the other hand, the systems that use gases (usually air) are called ***Pneumatic Systems***. They not only differ with the medium they use. Since there are certain differences in the properties of gases and liquids such as compressibility characteristics, design criteria's and components used in them differ. Some of graphic symbols used in drafting are also different.

2.2 Advantages of Fluid Power Systems

The power transmitting systems have both advantages and disadvantages[6].

The major advantages of the fluid power systems come from the versatility and manageability aspects. Fluid power systems are not restricted with the geometry and connections of the machine that it works on. Flexible pipes transmit power without being effected by the motions of the linkages on the system. Large force requirements of the systems can be handled without dealing strong components to transmit it as it is in mechanical transmission systems. Large forces can be controlled using smaller forces by simply pressing a button or pulling a lever with a small force. Obtaining infinitely variable speed for a rotary actuator is simple by using a fluid power system with a few simple components. A fluid power system can simply and efficiently multiply forces to obtain huge amounts as output. Constant force or torque demand of the system can be provided regardless of speed changes. Constructing of a fluid power system is simple and quick since all the elements required are standardized to some extend and readily available: no need to design and manufacture components as it is usually done in mechanical transmission systems. Automatic protection against overloads is a standard property of fluid power systems. Highest power per unit weight of the used system in power transmission can often be obtained by using fluid power systems. Maintenance cost can also be kept at a quite reasonable level.

2.3 Basic Hydraulic System

Figure 2.1 illustrates the basic structure and components of a typical hydraulic power system. Symbols used in the schematic are standard graphic representations of components.

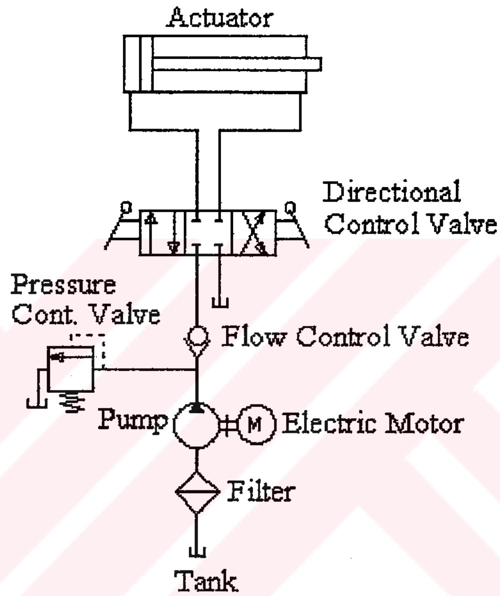

Figure 2.1 Basic hydraulic system.

As it is seen in Figure 2.1, a basic hydraulic system consists of components that are used in any system. A tank to store working fluid, a filter to remove particulate contaminants from the fluid, a pump to transmit mechanical energy into fluid, conductors that carry the fluid, a directional control valve to permit actuator to extend or retract by changing the path of pressurized fluid, a check valve (flow control valve) to prevent flow from the actuator to pump, a relief valve (pressure control valve) to prevent excessive pressures in the system. Any hydraulic system consists of such elements to perform the basic duties.

2.4 Components Of A Hydraulic Power System

Components of a hydraulic power system can be classified into four main items [7].

2.4.1 Energy Input And Transfer Devices

An electric motor or internal combustion engine is used as the prime mover in hydraulic circuits and the supplied mechanical energy is converted in to fluid energy by *pumps*.

Positive displacement pumps which supply a fixed flow rate to the system independent of the system pressure are normally used in hydraulic power systems. Flow rate may vary by a little amount due to internal leakage of the pump. Positive displacement pumps can be classified in to two subgroups: *Fixed Displacement Pumps* and *Variable Displacement Pumps*. Variable displacement pumps are designed so that as their flow rate can be changed.

If flow inlet and outlet directions can be changed, pumps are called *reversible pumps*, otherwise they are called *unidirectional pumps*.

Pumps do not "pump" pressure in to the system, they produce fluid flow. Pressure is determined by the resistance to this flow, as denied by the hydraulic system at the outlet of the pump.

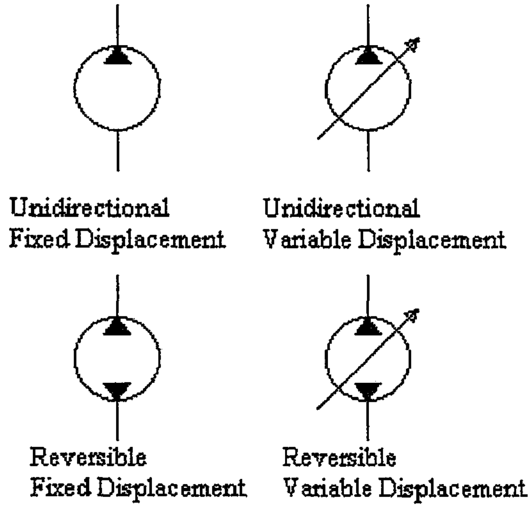

Figure 2.2 Graphic symbols for pump types.

2.4.2 Energy Modulation Devices

Energy modulation devices are used to control fluid flow and pressure by the use of devices called *valves*. Three basic types of valves exists:

i. Directional Control Valves: These determine the path thorough which the fluid flows. They can be directly controlled by the user (manually actuated) or by signals (pressure, electrical, or mechanical) given from some part of circuit due to pressure or actuator position changes. They are named according to the number of pipes they are connected (total number of input and output ports) for both input and output of flow and number of different positions they can be used (4 way 3 position valve, 4 way 2 position valve, 3 way 2 position valve etc.). A sliding spool in the valve change position, to change the path of flow through the valve, so inlet flow is directed to respective outlet ports. On graphical symbols, control types are presented at both end of valves, and the flow path configuration for each spool position is shown in a square.

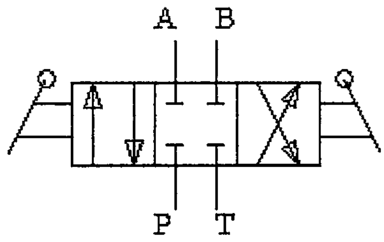

Figure 2.3 A manually actuated, 4 way 3 position valve.

For the directional control valve symbol in Figure 2.3, flow from or to any of the ports are not permitted for the current center spool position. For the left spool position, flow from port "P" is directed to port "A" and flow from port "B" is directed to port "T". Symbols on both sides of the valve represents a lever to manually control this valve. Different valve actuation types are available such as pilot operated (by pressure), solenoid operated, mechanically operated or spring centered.

In Figure 2.4 symbols of different types of valves are shown. Same symbols are attached on the valve itself by manufacturers.

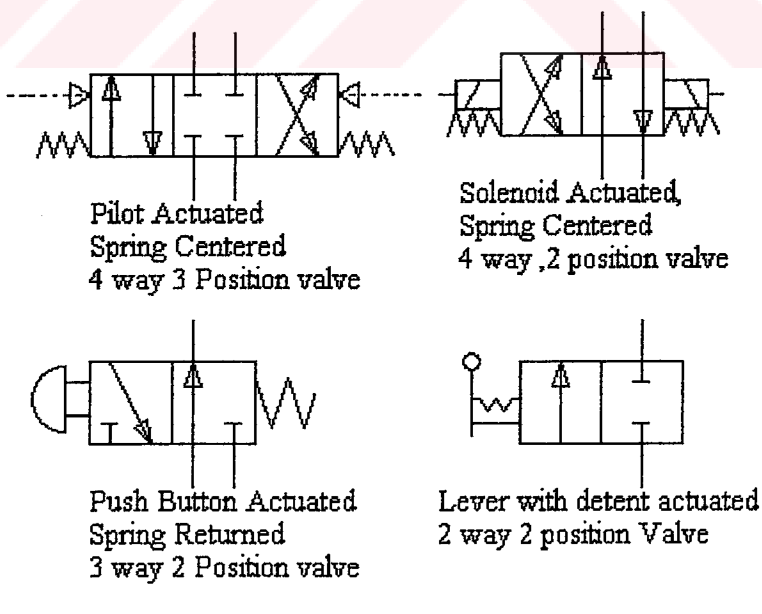

Figure 2.4 Different type of directional control valves.

ii. Pressure Control Valves: According to the pressure values set, they are opened or closed to change the path of the fluid. Most widely used type of pressure control valve is called a *pressure relief valve*. It protects the system against excessive pressure by opening and diverting pump flow directly to the tank at a certain set pressure. After the relief valve opened and flow is directed to the tank, pump runs against the relief valve's set pressure. Even though actuators do not operate, the system does work to send fluid to tank by passing the relief valve causing energy loss. Another typical pressure control valve is called a *sequencing valve* and when the pressure reaches its set value, it is opened and permits flow to be directed to different parts of the system by passing through itself. Pressure control valves take pressure values just from their inlet ports, or with a pilot line from any other point of the circuit. The second type of pressure control valves are called "pilot operated valves" since they are opened with the pressure controlled by the pilot lines. A valve called pressure reducing valve is used to maintain a lower constant pressure at its secondary outlet regardless of the pressure at its primary inlet and outlets. The symbols for the basic types of pressure control valves are shown in Figure 2.5.

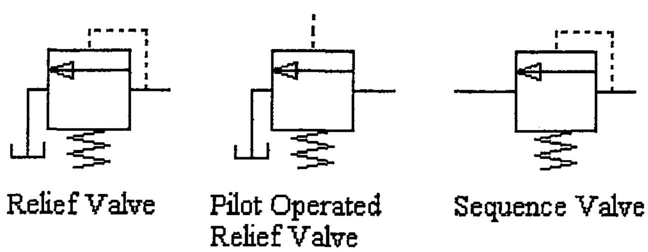

Figure 2.5. Frequently used pressure control valves.

iii. Flow Control Valves: They are used to control the speeds of the actuators in the system or prevent fluid flow in certain directions without dealing the pressure of the system. Widely used types are check valves and fixed or variable orifices. A check valve allows fluid passage only one direction through it and prevents fluid flow in the opposed direction (fluid flow from A to B is allowed but B to A is prevented for the check valve shown in Figure 2.6). Variable orifices are used to manipulate the pressure drop across thereby, when closed to a certain extent, may result in the opening of the relief valve reducing the flow rate going to an actuator.

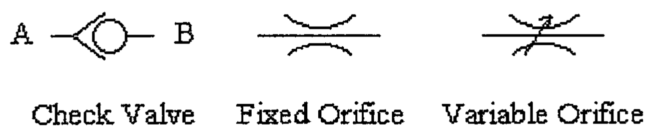

Figure 2.6 Most frequently used flow control valves.

Another type of check valve which is shown in Figure 2.7, called pilot operated check valve, permits flow from B to A if a pressure signal exist at port C.

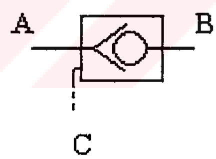

Figure 2.7 Pilot Operated Check Valve.

2.4.3 Actuators

The energy added to the hydraulic system is extracted from the fluid and converted into a mechanical output by components called actuators. Fluid power can be converted through linear motion by using *linear actuators* and rotary motion by using *rotary actuators*.

i. Linear Actuators (Hydraulic Cylinders): The pressurized fluid pushes the piston inside the cylinder. There are different types of hydraulic cylinders but operation principles do not differ. For *single acting* cylinders, fluid from the tank pushes the piston but they do not retract hydraulically. Piston retraction is accomplished by gravity, by a compression spring, or force exerted on the piston rod. Double-acting cylinders both extend and retract hydraulically. Some types of double-acting cylinders contain cylinder cushions at the ends of the cylinder to slow down the piston near the ends of the stroke to prevent impact when the piston is stopped by the end caps.

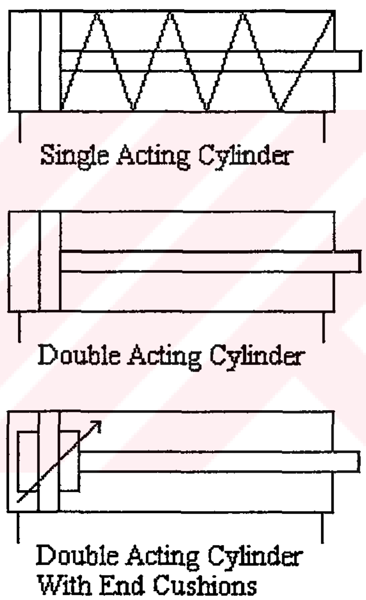

Figure 2.8 Symbols for different cylinder types.

ii Rotary Actuators: The mechanical power converted in to fluid power at pump is reconverted back into shaft power at rotary actuators. They can be classified in to two groups: continuous rotation and limited rotation. Continuous rotation actuators are called hydraulic motors. By a system with a pump and a hydraulic motor, infinitely variable speed control, self-overload protection, reverse rotation capability, and dynamic braking can be obtained.

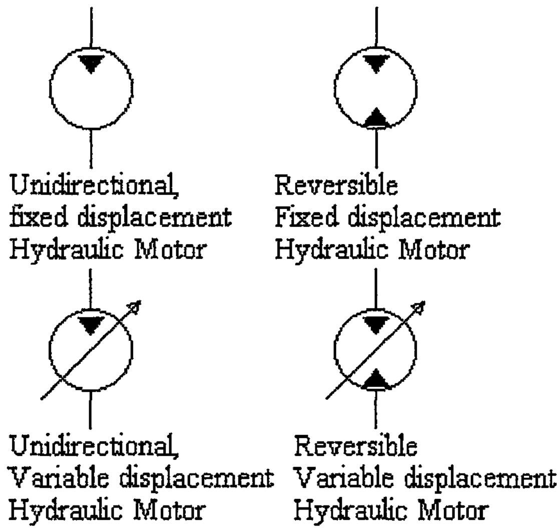

Figure 2.9 Hydraulic motor symbols.

2.4.4 Accessories

There are some other components used in hydraulic systems which are not included in the list of main elements. Some directly effect the working characteristic of the system, while some others are only used as auxiliary devices.

i. Accumulators: These are the devices that store the potential energy of an incompressible fluid held under pressure by an external. The accumulator can be used as a secondary power source.

ii. Filters: These are used to remove particulate contaminants from the working fluid. They can be located on section, pressure, or exit lines.

iii. Pressure gages and flowmeters: At any point of the system, fluid pressure and flow rate through pipes are measured using these instruments.

iv. Heaters, coolers, and thermometers: Temperature of the working fluid can be held in control by using these devices.

Symbols for commonly used accessories are given in Figure 2.10.

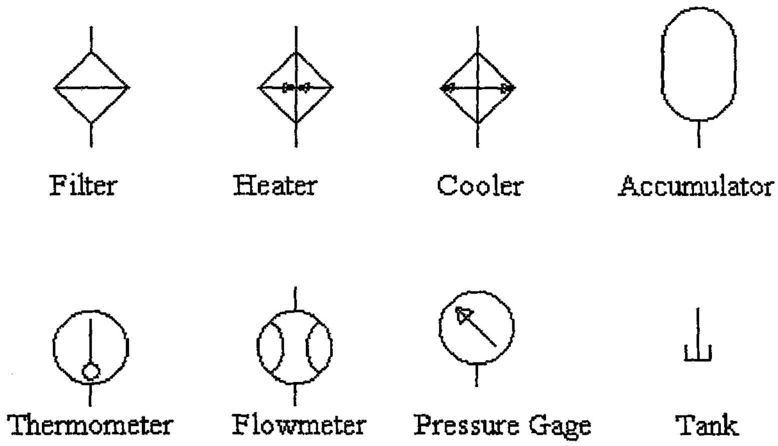

Figure 2.10. Commonly used accessories.

2.4.5 Fluid Conductors

In a fluid power system, the fluid is carried in the circuit by conductors. Mainly steel pipes, steel tubings, plastic tubings, and flexible hoses are used. Selection among them depends on the systems operating pressures and flow rates.

While some conductors in the system carry the pressurized fluid on the way from the pump to actuators, some others carry the fluid from outlets of the actuators, back to tank.

2.5 Types of Fluid Power Systems

Fluid power systems can be divided in to some major categories. At the beginning of this chapter, it is classified according to working medium as hydraulic or pneumatic. Another set of major categories of fluid power systems consists of open-loop and closed-loop systems.

2.5.1 Closed-Loop Systems

A closed-loop system uses feedback to match the desired output with the output value of the system in operation. The system output is sampled and to obtain desired value, the system input values are redetermined. If the velocity output of an actuator is less then the desired value because of fluid leakage, the flow rate into the system or desired portion of it, is increased to obtain desired velocity. Feedback transducers are used to control output and generally servo valves are controlled by signals coming from the transducers.

In Figure 2.11 a closed loop circuit, in which cylinder output is controlled and flow rate is readjusted, is shown [8].

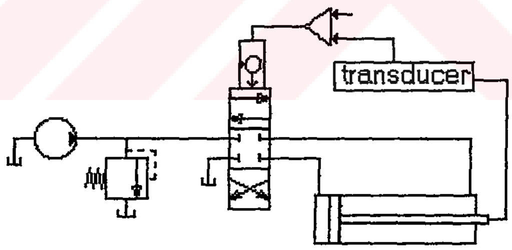

Figure 2.11 A closed-loop system.

2.5.2 Open-Loop Systems

In open-loop systems value of system output is not compared with the required design value. The output performance of the system depends on the characteristics of the components and the calibration of the system. Any errors of the system which effects the output such as leakage from actuator, change of fluid viscosity as the temperature changes are not compensated. No feedback is taken from the circuit. This type of hydraulic power systems are quite common in industry.

Open loop system can be classified further according to their function, method of control, and system type [7].

For another classification, whether the fluid returns the reservoir or directly reenters the pump is considered. The systems which the fluid flow is continuous, i.e. the fluid coming from actuator goes in to pump, are called ***Closed Circuit*** and the circuits in which the return lines goes into the tank are called ***Open Circuit***.

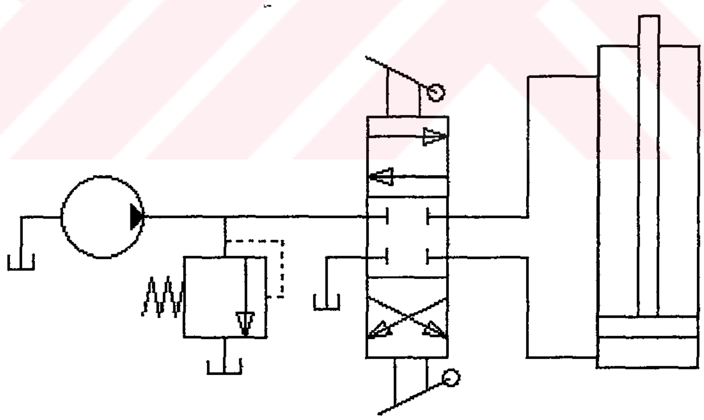

Figure 2.12 An open-loop open-circuit hydraulic system.

Only open-loop and open-circuit hydraulic systems have been considered within the scope of this thesis.

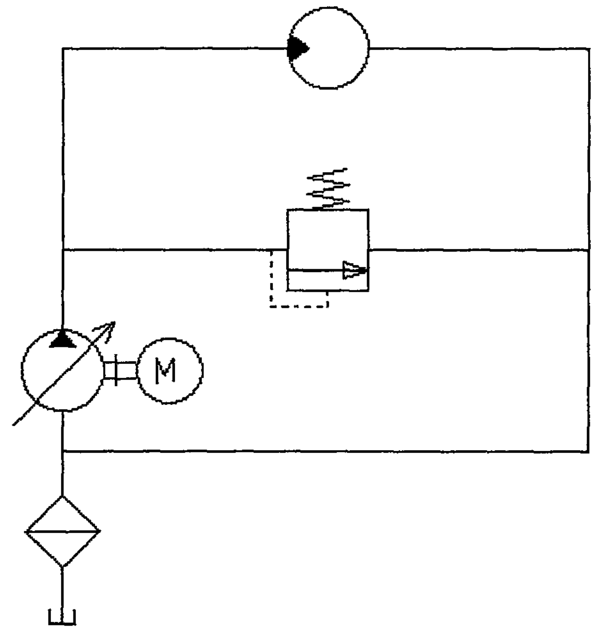

Figure 2.13 An open-loop, closed-circuit hydraulic system.

2.6 Hydraulic Power Applications

There are widely used classical fluid power systems to perform certain functions in industrial applications [7,8,9]. Components and the way they are connected can be duplicated and used in various different applications. Some of these interesting applications with their schematics and working principles are presented in the next section.

To understand how a circuit operates, different flowpaths with respect to positions of directional control valves or flow control valves are followed. In complicated circuits this may be quite difficult particularly when the load values effect the operation of pressure control valves, and in turn the motion of actuators. A software which simulates the operation of circuit by considering all parameters may be appropriate in such cases.

2.6.1 Single Cylinder Control Circuits

Simplest systems with the least number of elements are single cylinder control circuits.

i. Control of a single acting cylinder: Figure 2.14 shows a system with a single acting spring-return cylinder with a two way, three position, manually actuated directional control valve. For the spool position of the directional control valve shown in the figure, pump flow extends the cylinder. After the piston ends its stroke, pressure in the system increases and the relief valve is opened to send full pump flow to tank. For other spool position, pump flow again goes to tank through the relief valve, and the cylinder retracts with the force exerted by the spring.

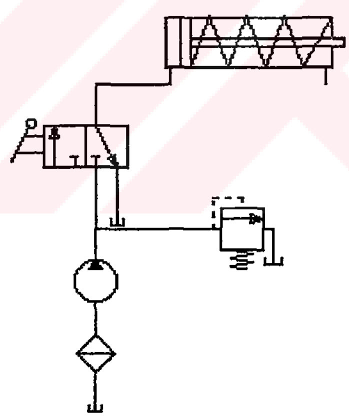

Figure 2.14 A single acting, spring returned cylinder circuit.

ii. Control of a double acting cylinder: Since the extension and the retraction of the cylinder is done by fluid flow, a four way valve is used. In Figure 2.15, when the four way, three position valve is in its center position the cylinder is locked and because of center spool type pump output directly send to tank, the pump is "*unloaded*". Pump is unloaded essentially at atmospheric pressure, so minimum power is consumed. When the left spool of directional control valve is activated, the pump flow pushes piston in forward direction. Since the valve is spring centered, at any time the lever is deactivated, center position of the valve prevails, the piston is locked at its current position. If the left directional control valve position is still active even after the piston reaches its end position, the system pressure increases and the pump flow goes to tank via the relief valve at set pressure of the relief valve. The right spool position is used to retract the piston.

At any time during the extension and retraction, if cylinder is overloaded, relief valve is opened and the system is protected.

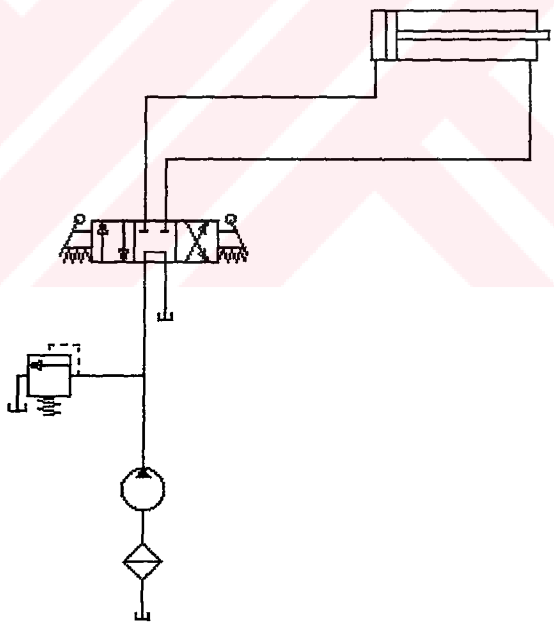

Figure 2.15 Control of a double acting cylinder.

2.6.2 Regenerative Circuit

Regenerative circuits are used to obtain increased extension speeds by joining the actuator exit by the inlet. Piston output is given to piston input for extension stroke, so extension speed is increased. This is a typical system used for drill applications. Left position of the directional control valve is used for slow feed, and center position is used for rapid advance without changing speed of pump shaft. When the center position is active fluid from the rod end of the piston regenerates the pump flow and this increases the flow rate entering other side of the cylinder. But since same pressure is applied at both sides of the piston, the load carrying capacity of the system is reduced.

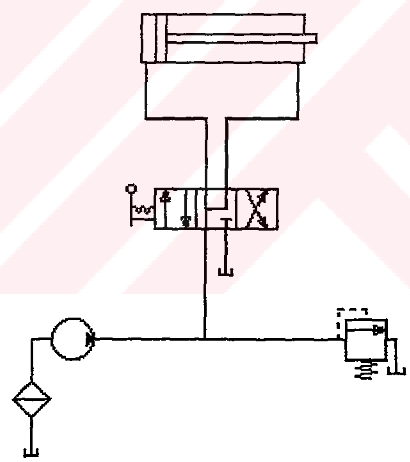

Figure 2.16 Regenerative Circuit.

2.6.3 Cylinder Synchronizing Circuit

When two cylinders are used in circuit, they can be either connected in series or in parallel.

i. Parallel connection of two cylinders: The two cylinders in Figure 2.17 is connected in series. Even if the cylinders are identical, if the loads on them are different, the piston with smaller load on it (i.e. the piston requires smaller pressure value to extend) moves first. After it ends its stroke, the system pressure rises and when the value is reached to push the other piston, the piston which requires higher pressure starts to extend.

In a hydraulic system *fluid always follows the easiest path*, i.e. the component (either an actuator or a pressure control valve) with smaller pressure demand is being activated first. In practice it is not possible to move two cylinders in synchronization connected in parallel even when the loads and cylinders are identical, because of different effects such as friction on the cylinders.

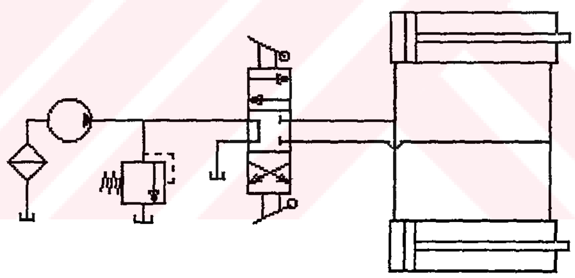

Figure 2.17 Two cylinders connected in parallel.

ii. Serial connection of two cylinders: To synchronize two cylinders, whatever the loads on them and their characteristics are, they can be connected in series. Fluid from the rod end of the first cylinder enters the blank end of second cylinder, and fluid from the rod end of second cylinder goes to tank. Thus two cylinders move in synchronization. Since the fluid volume at rod side of the first cylinder is used to push second cylinder by filling inside the blank end, piston area of second cylinder must be equal to the difference between the areas of piston and rod of first cylinder. Otherwise, after the first piston ends its stroke, second piston cannot continue on its motion or vice versa.

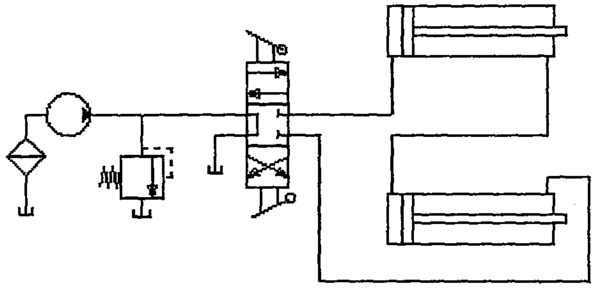

Figure 2.18 Cylinders connected in series will move in synchronization.

2.6.4 Cylinder Sequencing Circuit

Sequence valves are used for sequential operations in hydraulic circuits. The circuit in Figure 2.19, sequence valve 1 (PCV 1 on figure) is set a pressure value lower than that set for PCV 2. For extension stroke of cylinders, when the system pressure reaches the set value of PCV 1, it is opened and fluid passes through it, and pushes the first cylinder. After the first cylinder extracts fully, the system pressure increases and PCV 2 is opened, so the second piston extends. For retraction of the piston, there is no sequence, retraction is in random sequence. By changing the set values of sequence valves, extension orders of the cylinders are determined.

If the pressure required to push the first cylinder is greater then the set value of PCV 2, then PCV 2 is opened before the first piston extends, then sequencing order is determined only by pressure requirements of the pistons.

For retraction, fluid goes in to cylinders via the check valves connected in parallel with the sequence valves, so there will be no sequencing retraction.

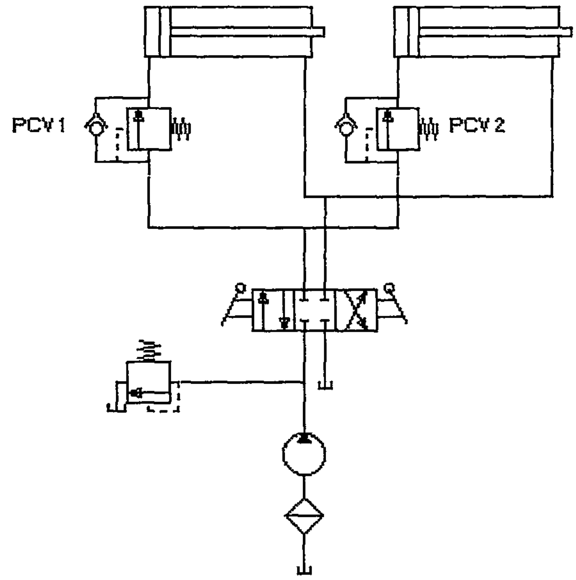

Figure 2.19 Sequencing Circuit.

2.6.5 Continuous Cylinder Reciprocating System

By using two sequence valves, pressure activated directional control valve, and pilot lines, the constructed circuit may produce continuous reciprocation of a hydraulic cylinder. Such a circuit is shown in Figure 2.20. Each of the sequence valves senses a stroke completion by the corresponding build up of pressure. When the sequence valve is opened, a signal is send to pressure control valve to shift it to other position. Thus continuous reciprocation of the actuator piston is obtained.

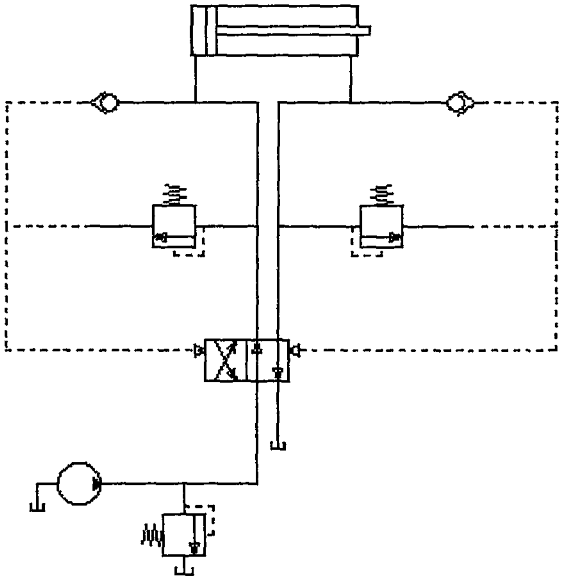

Figure 2.20 Continuous cylinder reciprocating system.

2.6.6 Hydraulic Motor Speed Control

Infinitely variable speed control can be obtained for fluid power systems, by using a variable orifice and a relief valve in the system. For the system in Figure 2.21, speed of the hydraulic motor can be varied by adjusting the setting of the variable orifice. This increases the pressure in the branch resulting in the relief valve opening and some amount of oil goes through the relief valve. Then, change in the flow rate into hydraulic motor is reflected in change of shaft speed.

Velocity of linear actuators can also be controlled with the same system. Figure 2.22 shows a cylinder velocity control system. If the velocity of the cylinder is controlled with the orifice placed at the inlet line, which is controlling flow rate in to the cylinder, is called meter in circuit. If the speed control is obtained via the orifice placed on the line going to tank from cylinder outlet, which increases the backpressure of the cylinder, the circuit is called meter-out circuit.

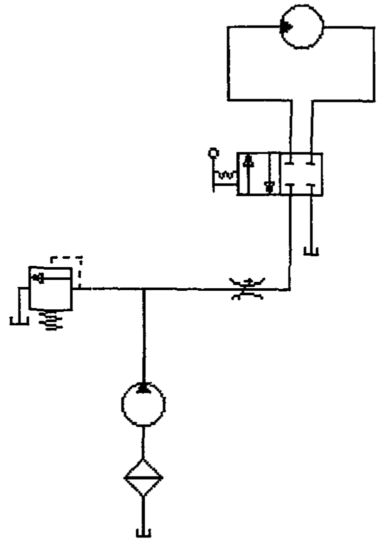

Figure 2.21 Speed control of a hydraulic motor.

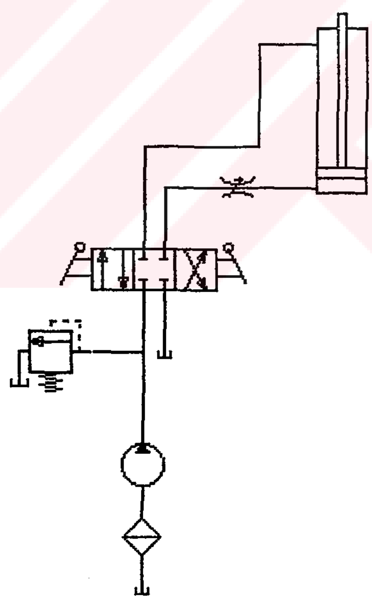

Figure 2.22 Meter-in circuit.

2.6.7 Accumulator Circuits

Accumulators can be used in a hydraulic system as an auxiliary or primary power source, leakage compensator, or hydraulic shock absorber. Size of the accumulator and precharge pressure is calculated according to system's operating conditions.

Accumulator is used as the primary power source in the circuit shown in Figure 2.23. The pump flow fills the accumulator after the system is blocked by closing directional control valve or cylinder ends its stroke.When a certain pressure is reached, pump is unloaded by the pilot operated relief valve (unloading valve). After directional control valve shifts a position to move the cylinder, fluid flow to the system is supplied only by the accumulator. If the pressure of the accumulator drops below a certain value during cylinder operation, unloading valve is closed and the pump feeds the system.

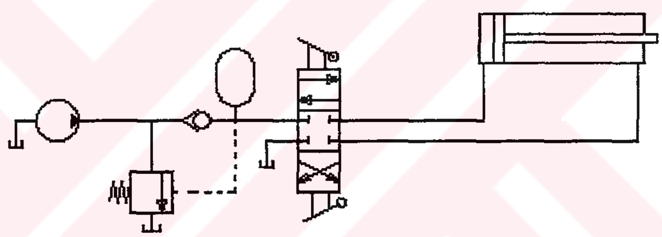

Figure 2.23 Typical accumulator circuit.

CHAPTER 3

THE COMPUTER PROGRAM

In this chapter, development of the computer program, its capabilities, usage, methods, and algorithms used for every facility are explained in detail.

3.1 Development Environment of the Program

DraftSima is written by using a powerful software development tool; Microsoft Visual Basic v3.0 Professional Edition. It runs on Windows operating environment.

Program development procedure in Visual Basic consists of two basic stages: creating the user interface, and writing the code to work.

User interface includes the forms and controls. A *form* is a window that makes up part of an application's interface. A *control* is an object that resides on a form to enable user interaction with the application. Controls accept user input or display output. Both the forms and the controls are named as *object*. Objects have properties that define aspects of their appearance, such as position, size, and color, and aspects of their behavior, such as how they respond to user input. No code is necessary to adjust properties of the controls but if required, properties of the controls can be changed by code while the program is running. Designing user interface;

creating menus, placing command buttons or different types of controls are all easy and not time consuming. Programmer spares minimum time for user interface, and has more time for developing code for special applications.

Visual Basic is an *event-driven programming language*. Event is an action recognized by an object, such as pressing the mouse button or a key, and for which the code is written to respond. Events can occur as a result of user action or program code. If some part of the code is desired to execute when the user clicks on the control, the code is written in "click" event of the relevant control.

DraftSima includes different forms and controls. For every group of hydraulic circuit components, different forms exist and user can select the components by clicking the relevant control placed on the form.

Program has a modular structure, and suitable for adding new features and capabilities without changing the existing code, by simply adding new code.

3.2 Capabilities of the Program

DraftSima has the following capabilities:

1. Most of the hydraulic circuit components, a designer would to use in most typical applications, are included in the component library of the program. Graphic symbol for each component conforms the standards ISO 1219 and TS 1306.

2. All the components are classified under the headings according to their main functions. For each heading there is a different form, and desired component symbol can be selected visually.

3. Selected component can be placed anywhere on the screen by dragging with the mouse. There are no limitation on the number of elements used and the size of the circuit schematic.

4. Orientation of any component on the screen can be adjusted. Components can be rotated, or mirrored about the vertical or the horizontal axis. In a hydraulic circuit schematics, a component can be used in any orientation to expand the schematic in desired direction. So any circuit schematic previously drawn on paper can be constructed on the screen.

5. Components can be removed from the system by simply selecting them and pressing "delete" key on keyboard.

6. Component can be connected by lines. Starting and ending points of lines are placed by snapping grids. So lines from or to the connection ports of elements can be drawn precisely without any special effort. Since only orthogonal lines are used to represent conductors in hydraulic circuit schematics, drawing lines are constrained to be horizontal or vertical. Pilot lines can also be drawn.

7. Comment texts can be attached to schematics. Explanatory comments, and notes can be written anywhere on the circuit and can be relocated at any time.

8. With zoom facility, visual image of the drawing on screen can be magnified or shrunk. When the display is *zoomed out*, a large portion of the drawing can be seen. *Zooming in* can blow up a small portion of the drawing, and show more of its details. When working with the large schematics, zoom in and zoom out facility of the program makes drafting easier.

9. Schematics can be printed on paper. Heading, drawing number, and date can be added by using a title block facility.

10. Constructed systems can be saved and kept on disk files with all relevant data entered for the components.

11. Open-loop, open-circuit hydraulic systems drawn in the program can be simulated by animating the functions of components.

12. Speed of simulation can be adjusted.

13. Simulation can be observed step by step. Program waits for a user input to continue after animation of each component. This is a helpful facility for following the operation behavior of complex circuits.

14. There are no limitations on number of components for animation. All the actuators that must work together can be animated together whatever the total number of actuators may be.

15. Program responds to user events at any time during simulation. At any step of an actuator movement, by changing the position of directional control valve, actuator can be stopped at its recent position.

16. It is possible to observe pressures and flow rates.

3.3 Limitations of the Program

1. Selection of hydraulic power components is limited by the component library. A new element type can not be added by the user at run-time of the program. On the other hand source code can be simply modified for adding a new component.

2. Software is limited to the simulation of the steady state operation of open-loop, open-circuit systems. Closed-loop circuits, and closed-circuit systems can not be simulated. Transient behavior of the circuits can not be simulated either.

3. Operation of multidirectional pumps can be simulated only in one direction, since closed circuits cannot be simulated.

3.4 Main Program Screen

Main screen of the program consists three main parts. *Menu bar* where all commands can be selected, *drawing area* where schematics are constructed, and a *toolbar* which can be used as a short-cut for frequently used menu items. A status bar is also placed at the bottom of the screen to show current filename.

Drawing area has a horizontal scroll bar placed at the bottom, and a vertical scroll bar placed at the right side of it. Scroll bars are used to pan the image on the drawing area without changing the magnification.

Using the buttons on the toolbar, frequently used commands for component selection, zooming, printing or standard file operations can be reached.

Main screen of the DraftSima is seen in Figure 3.1.

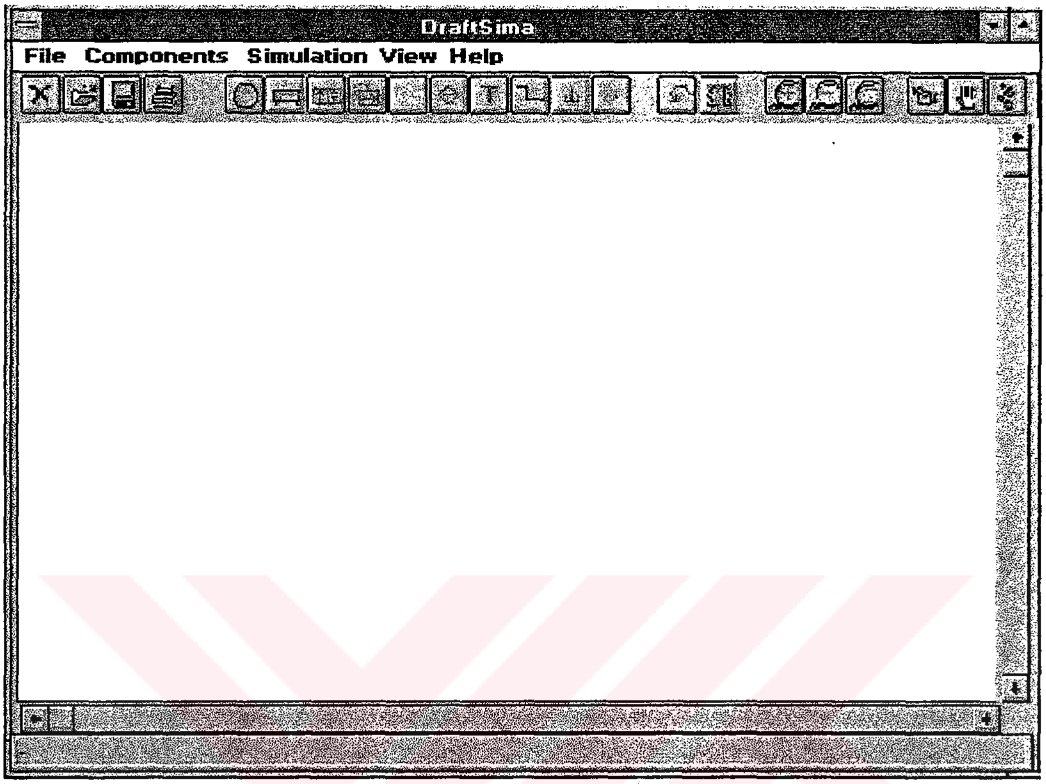

Figure 3.1 Main screen of the program.

3.5 Programming Procedures: Drafting.

In this section, the facilities available for drafting the schematics and special algorithms used for drafting modules are explained in detail.

3.5.1 Drafting The Schematics

Component library is classified under five headings. For each component group, a form is designed to select the components. The forms can be reached from the “Components” menu, or by using the buttons on the toolbar which have pictures on them showing the symbol representing the relevant group. Items included in the "Components" menu are seen in Figure 3.2. Buttons on the toolbar related with component selection are shown in Figure 3.3.

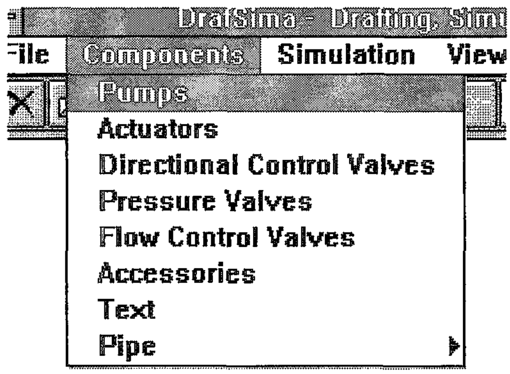

Figure 3.2 Components menu of DraftSima.

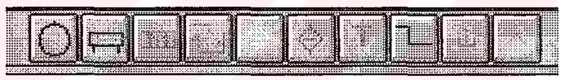

Figure 3.3 Buttons on the toolbar related to component selection: Pumps, actuators, DCV's, PCV's, FCV's, accessories, text, pipe, tank buttons

i. <u>Pump Selection Form</u>: Different pump types can be selected from the "Pumps" form, reached by selecting the "Pumps" menu item or the first button shown on Figure 3.3, which have a pump symbol on it. Figure 3.4 shows the "Pumps" form.

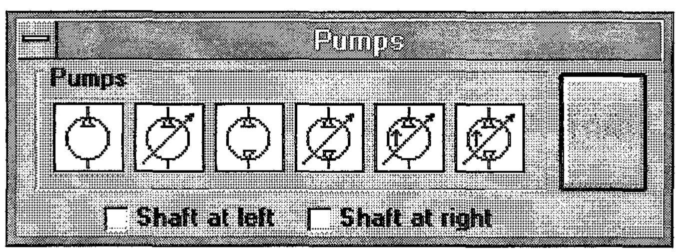

Figure 3.4 Pumps form.

On the form, different pump types are drawn on white boxes. Desired type can be selected by clicking the mouse on it. The red rectangle around the white box shows the selected one. By clicking on the "LOAD" button or pressing the ALT+L key combination loads the components on the drawing area. The loaded component moves with the mouse on drawing area and can be placed to desired location by pressing the *left mouse button*. The available pump types are :

1) Unidirectional pump, fixed displacement
2) Unidirectional pump, variable displacement
3) Multidirectional pump, fixed displacement
4) Multidirectional pump, variable displacement
5) Unidirectional pump, variable displacement, pressure compensated
6) Multidirectional pump, variable displacement, pressure compensated

The form contains two check boxes with the titles "Shaft at left" and "Shaft at right". If shaft connections are desired on the pump symbol for representing the electric motor or internal combustion engine connection on any side of the pump, one or both of the check boxes can be marked.

ii Actuator Selection Form: Both the linear and rotary actuators can be selected from "Actuators" form. Five linear actuators (cylinders) placed on the first page. The arrow button is used to go to second page where three more linear actuators can be found. The "Actuators" form is shown in Figure 3.5.

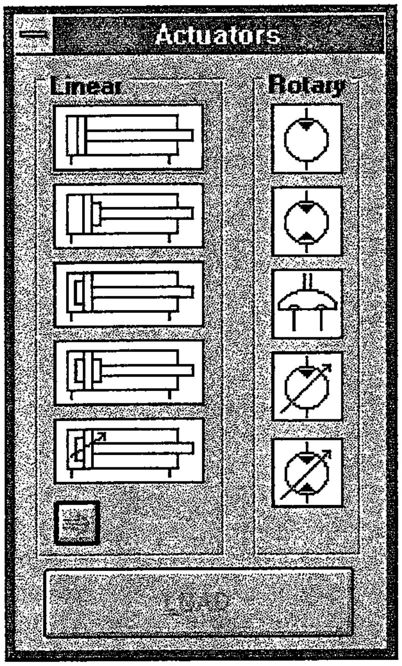

Figure 3.5 Actuator selection form.

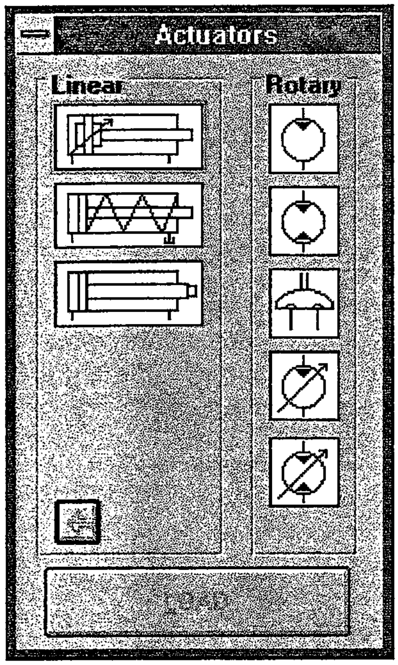

Figure 3.6 Actuator selection form.

Now the arrow button is used to turn the first page. Available linear actuators, in the order they reside on form, are

1) Double acting cylinder

2) Double acting cylinder with cushion on rod side

3) Double acting cylinder with cushion

4) Double acting cylinder with cushion at both ends

5) Double acting cylinder with adjustable cushion on one side

6) Double acting cylinder with adjustable cushion on both sides

7) Single acting cylinder, spring returned

8) Telescopic cylinder

And the rotary actuators are:

1) Hydraulic motor, one sided, fixed displacement

2) Hydraulic motor, double sided, fixed displacement

3) Semi-rotary actuator

4) Hydraulic motor, double sided, fixed displacement

5) Hydraulic motor, double sided, variable displacement

iii. Directional Control Valve Selection Form: There are many alternatives for directional control valves. Any desired valve combination can be constructed by selecting the number of ways, number of positions, any spool type for any position, and for each end of the valve one of the available control types. After constructing the valve, it can be placed in to the drawing by clicking the "LOAD" button.

Directional control valve selection form is shown in Figure 3.7. Number of ways can be selected from the combo box next to the "Ways" label and number of positions can be selected from the combo box next to the "Positions" label. After selecting number of ways and number of positions, an empty directional control valve symbol is drawn on the white area of the form. Available symbols representing different spool positions are shown on a frame at the right of the screen according to the selected way number. After selecting any one of the spool types from the "Spool Types" frame, by clicking on one of the rectangles in empty directional control valve symbol, selected spool type is assigned to this position of the valve. Spool types for other positions are determined in the same way. 12 alternatives for four way valves, 3 alternatives for three way valves and 2 alternative for two way valves exist. Available control types are given on a frame titled "Actuator". Selected actuation type is assigned to the directional control valve by clicking on left or right side of the symbol. At any time, three types of actuators are shown on the frame, and using the arrow button, different actuators can be reached. Total 16 different actuator types are available. In Figure 3.7, initial view of the directional control valve selection form is seen. The first item in "Spool Types" frame and the first item in "Actuator" frame are currently selected by default.

Directional control valve selection form can be reached by clicking "Directional Control Valves" item from "Components" menu, or, by clicking the button on the toolbar with the directional control valve picture.

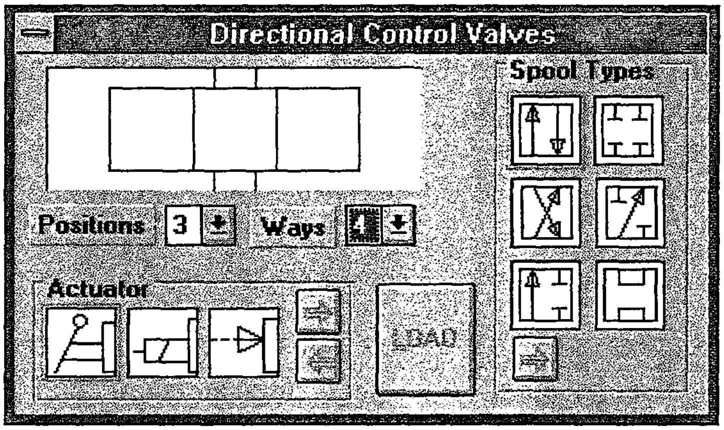

Figure 3.7 Directional control valve selection form.

A four way, three position directional control valve constructed by selecting different spool types for each position and lever actuator for both ends is shown in Figure 3.8. Another valve configuration, a three way, two position valve with lever actuation at left and spring at right, is shown in Figure 3.9.

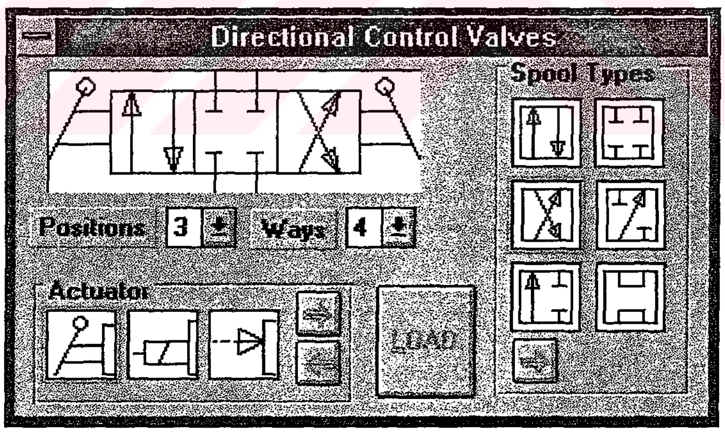

Figure 3.8. A lever actuated, four way, three position valve.

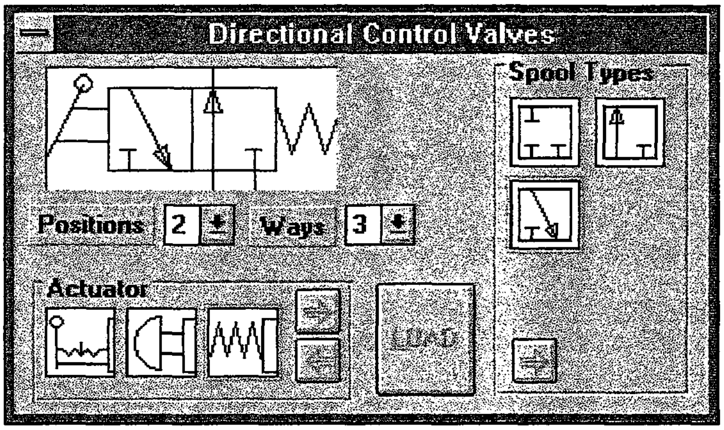

Figure 3.9. A lever actuated, spring returned, three way, two position valve.

iv. Pressure Control Valve Selection Form: Pressure control valves can be selected from the "Pressure Control Valves" form. Nine different types of pressure control valves included in the form. Figure 3.10 shows the "Pressure Control Valves" form. This form can be opened by clicking the "Pressure Control Valves" item from the components menu, or by clicking the toolbar button with the pressure control valve picture. When the form is opened the first valve is selected by default. The user needs to click on the desired valve and then on the "LOAD" button to place the valve on the main drawing screen. Available pressure control valves are:

1) Relief valve
2) Pilot operated relief valve (unloading valve)
3) Sequence valve
4) Pressure reducing valve
5) Counterbalance valve
6) Piloted relief valve
7) Piloted sequence valve
8) Counterbalance valve with check
9) Sequence valve with check

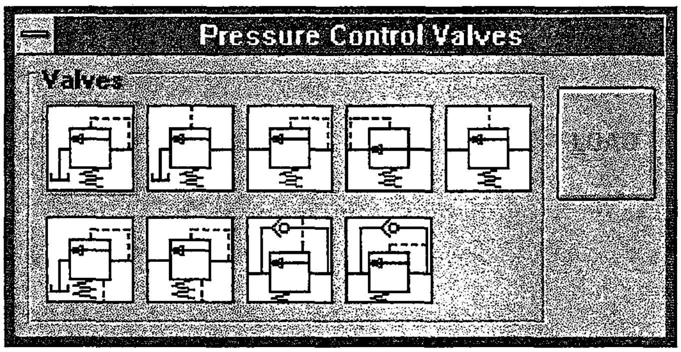

Figure 3.10 Pressure control valve selection form.

v. Flow Control Valve Selection Form: Flow control valves can be selected from the "Flow Control Valves" form. Ten different types of flow control valves included in the form. Figure 3.11 shows the "Flow Control Valves" form. This form can be opened by clicking the "Flow Control Valves" item from the components menu, or by clicking the toolbar button with the flow control valve picture. When the form is opened the first valve is selected by default. The user needs to click on the desired valve and then on the "LOAD" button to place the valve on the main drawing screen. Available flow control valve are:

1) Check valve
2) Pilot operated check valve
3) Fixed Orifice
4) Variable (adjustable) orifice
5) Shuttle valve
6) Pressure compensated flow control valve
7) Pressure compensated flow control valve for both directions
8) Pressure compensated flow control valve with bypass
9) Orifice with check valve
10) Variable orifice with check valve

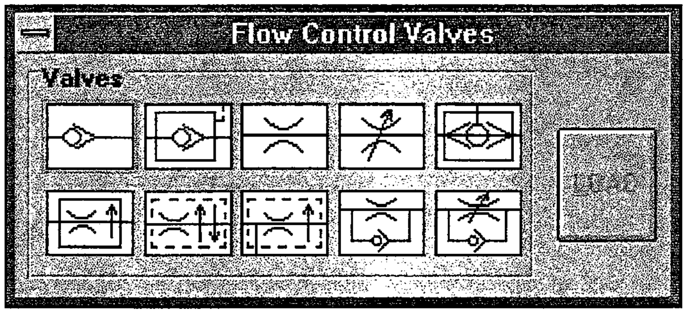

Figure 3.11 Flow control valve selection form.

vi. Accessories Selection Form: Accessories can be selected from the "Accessories" form. Nine different types of accessories included in the form. Figure 3.12 shows "Accessories" form. This form can be opened by clicking the "Accessories" item from the components menu, or by clicking the toolbar button with the *filter* picture. When the form is opened the first item is selected by default. The user needs to click on the desired one and then on the "LOAD" button to place the component on the main drawing screen. Available accessories are:

1) Accumulator
2) Filter
3) Heater
4) Cooler
5) Thermometer
6) Pressure gage
7) Flowmeter
8) Electric motor
9) Internal combustion engine

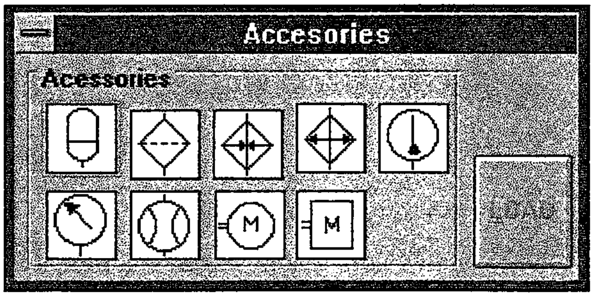

Figure 3.12 Accessory selection form.

vii. Drawing Pipes: To start drawing a pressure line, first "Pipe" item from "Components" menu is clicked. Pipe drawing procedure then begins by selecting the "Pressure Line" item from the sub menu. Instead of using this menu item, Ctrl+Z key combination or the pipe button on the toolbar can be used. Starting point of the line can be given by pressing the *left mouse* button, then *without pressing any of the mouse buttons*, cursor is moved on the screen to a location either in horizontal or vertical direction. End point of line is given again by pressing the left mouse button. If no more segments is desired, by pressing the *right mouse button*, drawing operation is completed. End points of lines are snapped to certain locations, a spacing of five "DraftSima units" is used for snapping, so end point of other lines or connection points of component symbols can be snapped without any extra effort.

viii. Adding Text To Schematics: Texts can be added to the circuits by using "Text Editor" utility of the program. Text editor can be reached by selecting "Text" item from "Components" menu or clicking the toolbar button with "T" on it. Text is entered to the part of the form labeled as "Enter Text". If two lines of text is desired, "enter" key is pressed. Without pressing enter, even if the text continue from the second line because of size of the box, the resulting text will be a single line on the screen. From the list box labeled as "Available Fonts", one of the fonts supplied by the system can be selected. Font size can be adjusted by using up or down arrow buttons placed under the fonts list. Font properties such as italic or underlined, can be adjusted by selecting the items in the frame labeled "Properties". Appearance of the text on the drawing can be checked from the part "Sample Text" before loading it. Figure 3.13 shows the text editor , and a sample text written on it. After the text is placed on the main drawing screen, its position can be changed at any time as it can be done for every component. Selecting, moving and rotating the components are explained in the next section.

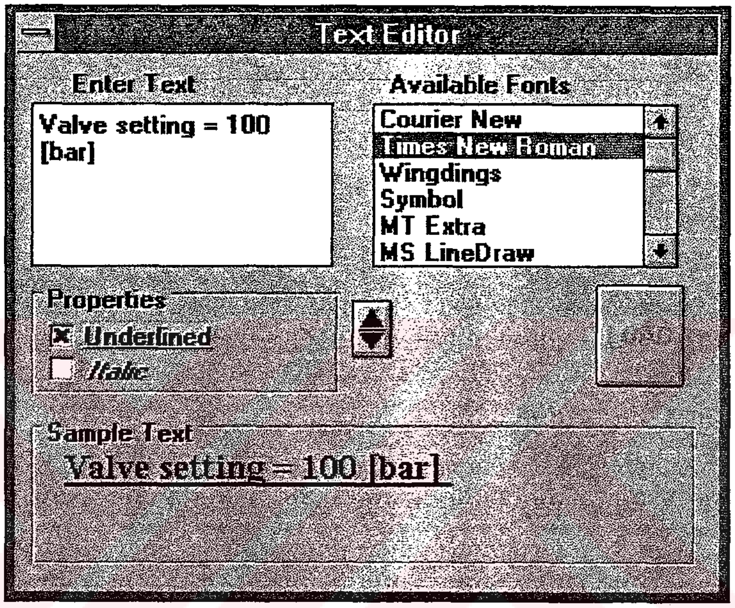

Figure 3.13 Text editor form.

3.5.2 Editing The Schematics Drafted

Positions and orientations of components on the drawing can be changed. To move a component, *left mouse button* is pressed when the cursor is on the component and by holding the button pressed , component is dragged to its new location and mouse button is released. To rotate or mirror a component, it must be first selected by pressing on it with the *left mouse button* while the "*Control*" key is pressed. Then the color of the selected component turns in to red. Using the rotate button from the toolbar, the selected component can be rotated. Instead of using rotate button, *Control+Left arrow* key combination can be used to rotate elements by 90° in counterclockwise direction and *Control+Right arrow* key combination can be used to rotate elements by 90° in counterclockwise direction. To take mirror image, mirror button on the toolbar is used. To delete the component, “delete” key is pressed after it is selected.

Texts can be moved, selected and deleted in the same way with the components. But texts cannot be rotated or mirrored.

Lines can be selected by clicking on them without pressing any other key. Pipes cannot be moved, rotated, or mirrored, but can be deleted.

In Figure 3.14, rotate and mirror buttons of the toolbar is shown. Eight different orientations of a pressure control valve obtained by rotate facility is seen on Figure 3.15.

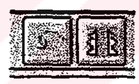

Figure 3.14 Rotate and mirror buttons on toolbar.

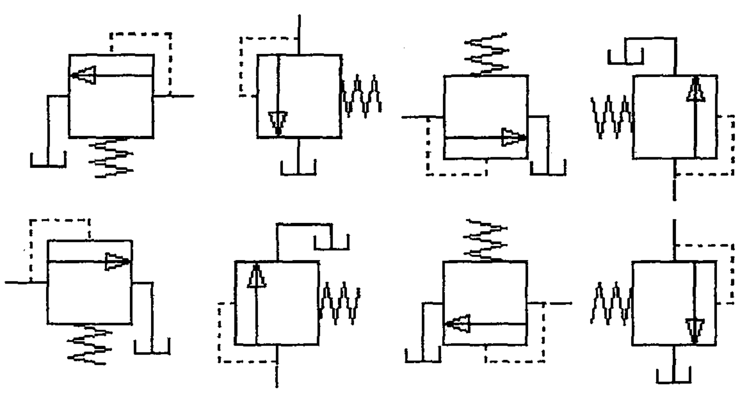

Figure 3.15 Eight different orientation of a pressure

3.5.3 Zoom Facility

With the zoom facility, visual image of the drawing on the screen can be magnified or shrunk. The commands related with zoom facility can be selected from "View" menu, or from the buttons placed on toolbar. When "Zoom Out" is selected, the visual image is shrunk by a scale factor of two. When "Zoom in" is selected the visual image of the screen is magnified by a scale factor of two. Upper left corner of the screen remains unchanged after "zoom in" and "zoom out" commands. For "zooming in" desired portion of the screen by specifying upper left corner and lower right corner, "Zoom Window" facility is used. For turning the initial view of the screen, "Initial Page" item selected from the "View" menu.

As a drawing aid, grids can be used for drawing pipes. Even when the grids are off, pipes are still drawn by snapping special coordinates.

3.5.4 Print Utility

A print facility form is designed to adjust printing options. Figure 3.16 shows the form designed for print options.

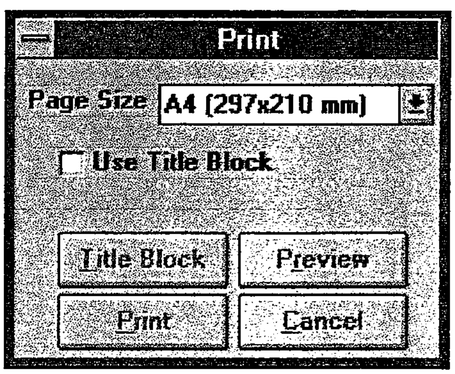

Figure 3.16 Print facility form.

Page size can be selected from the combo box labeled as "Page size". Depending on the system used, schematics can be printed on A4 size or A3 size paper.

A title block can be attached to print-outs. Fields of the title block is first filled, and by putting a check to the check box labeled "Use Title Block", it is attached to print out. By pressing the "Enter" key, moving between fields of title block is possible. Title block design form can be reached by pressing the "Title Block" button. In Figure 3.17 a sample filled "Title Block" is shown.

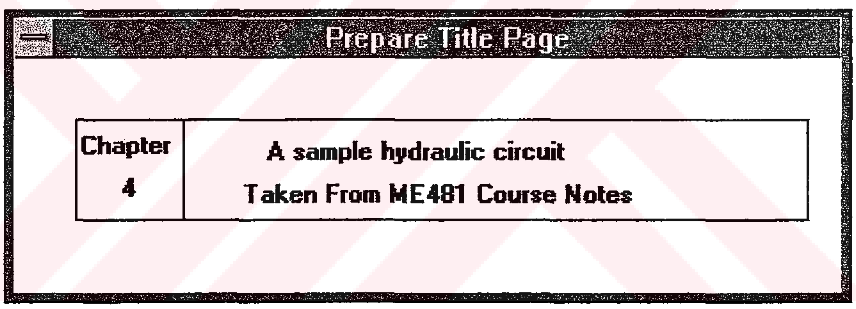

Figure 3.17 Filled title block design form.

The "Preview" key is used to control the way schematics printed on paper. The white base of the "Print Preview" form represents the limits of the paper. The same image seen on the preview form is printed on paper. Figure 3.18 shows the preview form with a sample circuit.

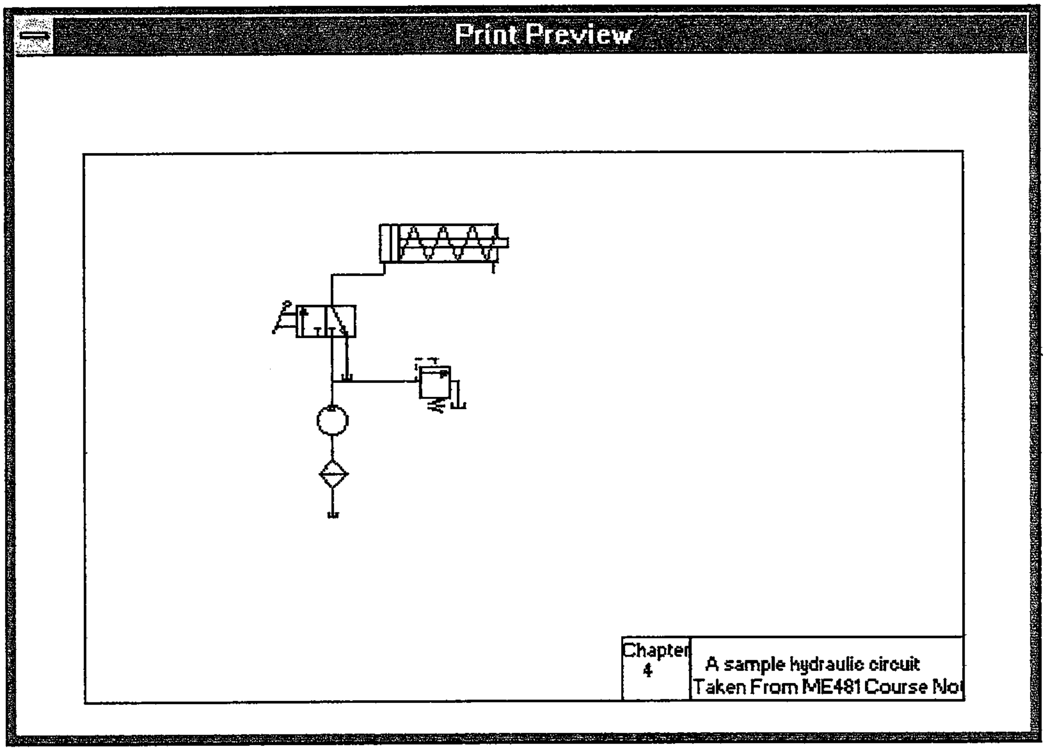

Figure 3.18 Print preview page.

Print command works with the principle "What you see on screen is what you get on paper". So any desired portion of schematics can be printed on paper, first by using the zoom facility to see it on screen.

3.5.5 Procedures Applied For Drafting Utilities

The program has a part called *drawbase* which behaves as the container of the schematics. *Drawbase* is created by using a Visual Basic object type, "picture box". Picture boxes are also used for drawing image of each component.

Picture box is a control which displays a picture either assigned from a *graphic file* or drawn on it by *graphic methods*. Graphic methods perform run-time drawing operations either on forms or on picture boxes. Graphics methods are line, circle, point, cls, pset, and print. Line is used to draw a line between two specified points. Circle is used to draw a circle or an arc with a given radius, center point and if necessary start and end angles. Cls is used to clear the image on the picture box or form. To display a picture on a picture box, one of the two methods can be used. The graphic image can be loaded from an external graphic file previously drawn with another drawing program. Graphics methods can also be used.

Loading external files as the source of an image has certain disadvantages. First of all, for a software like DraftSima, approximately 400 graphics files should be available every time to run the program. This increases the size of the program files and space occupied on disk. If some of them are damaged, program cannot be run. Another major disadvantage is the usage of operating system's graphical memory sources reserved for user programs. Loading picture files rapidly reduces the "user resources" of the Windows, and maximum number of components that can be used for such a program becomes limited to construct complex circuits.

DraftSima uses graphic methods to draw images of component symbols. For each component a picture box is used and graphic image of the component is drawn with line, circle and pset methods on picture boxes.

Picture boxes have an important property, Autoredraw, which determines the reserved memory for the graphical image carried. If this property is set to *true*, graphics drawn on the picture box is stored as a persistent bitmap in memory. The stored picture is redrawn when certain kinds of screen events occur, for example, when a form is redisplayed after being hidden behind another window. Setting the property *true* for picture boxes causes certain graphical memory problems when the amount of picture boxes (i.e. components for DraftSima) increases. If Autoredraw property is set to *false*, automatic repainting of an object is disabled and graphics output is written only to the screen. This saves important amount of memory, and eliminates the limitation on number of picture boxes and on the components.

Visual Basic invokes the object's *Paint event* when necessary to repaint the object. The code written under *Paint event* of the object is executed, instead of redrawing object. For the desired graphic output, necessary code is written in *Paint event* of the picture box.

For each component group, such as directional control valves, flow control valves; at design time, a picture box is placed on drawbase. To load many components of the same type, control arrays are constructed for each group of components. A control array is made up of a group of controls of the same type that share a common control name and a set of event procedures. Each control in the array has a unique index number. All items in a control array must have the same name property. The names of picture boxes making up the control array are, *pump* for pumps, *piston* for hydraulic cylinders, *fcv* for flow control valves, *pcv* for pressure control valves, *motor* for hydraulic motors, and *acces* for accessories. There are also arrays for storing numbers to hold properties and types of components used. They are named as *pumparray* for pumps, *pistonarray* for hydraulic cylinders, *fcvarray* for flow control valves, *pcvarray* for pressure control valves, *motorarray* for hydraulic motors, and *accesarray* for accessories.

When a component selected from one of the component selection forms, a picture box is loaded as the new element of the related component's control array. A variable is increased by one and total number of elements of this component is controlled by checking this variable. Then type of the component is written to in the component's array. After the picture box loaded, paint even is invoked, and selected component is drawn on the picture box.

As an example, after a directional control valve is constructed on "Directional Control Valves" form, and "Load" key on the form is pressed, the following procedure applies:

1. The variable which holds number of directional control valves loaded, *ndcv*, is increased by one.

2. Number of ways, number of positions, each selected spool type, selected actuation types for both sides, orientation of the valve symbol is written in a row of the *dcvarray*.

3. *ndcv*'th component of the picturebox array, named *dcv*, is loaded (a new control appears on screen), and moves on the screen as attached to the cursor.

4. By clicking the left mouse button, picture box is placed on the *drawbase*.

5. Paint event of the picture box named *dcv(ndcv)* is invoked, and by checking the values stored in dcvarray with if..then..else.. statements of the programming language, graphics of the valve symbol is drawn.

When position of an element on the screen is desired to be changed, cursor is located on the element and left mouse button is pressed. There is no need to write code to determine which element is selected, because when the mouse button is pressed on the picture box, *mouse down* event-procedure is invoked automatically by the system and a variable, *index*, returns the value of the selected component's control number.

There are two procedures named as *resize* and *relocate* to arrange the size and positions of controls after zooming or panning with control boxes.

When the zoom facilities of the program are used, scaling of the drawbase is changed, and all the picture boxes resized by *resize* procedure. Width and height values of picture boxes containing component symbols does not change. Instead of changing dimension values of each element, only the scale of drawbase changes. .Initially, coordinates of the upper left corner of the *drawbase* is X=0 and Y=0 and lower right corner of the drawbase is X=600 and Y=350 in pixel units. When the zoom in command is used, scaling of the *drawbase* changes, X and Y values of lower right corner is divided by two. Then *resize* procedure gives every picture box placed on drawbase to their original dimensions. When the width of drawbase is 600 units by 350 units, size of a four way, three position directional control valve is 90 units by 30 units. After scale of drawbase is changed to 300 units by 175 units, a 90 units by 30 units picture box appears twice as the previous screen. This method, changing scale of *drawbase* instead of with and height values of every picture box, makes zoom commands run much faster.

Of course, after changing the scale of the *drawbase*, position of each picture box on *drawbase* changes, but their coordinate values in X and Y remains unchanged. With the initial scale of the *drawbase*, coordinates of the center of screen is X=300 and Y=175 point. A picture box initially placed at center of screen from its upper left corner, has coordinates recorded in array as (300,175). After zooming in, when the X=300 and Y=175 coordinate becomes lower right corner of the drawbase, the coordinate values of picture box remains unchanged, but it is carried to new location of (300,175) point. Since coordinate values of all picture boxes are remain unchanged, and only the coordinates of *drawbase* changes, computational time decreases significantly. This makes the operation of DraftSima much faster.

Mouse_Down is an event procedure of Visual Basic objects. The code written in *mouse_down* event of any object, is executed when any of the buttons are pressed when the cursor is positioned on this object. The variable *button* automatically returns a value to indicate which of the mouse buttons are pressed. The two variables X and Y returns the coordinates of the cursor at the moment the mouse button is pressed. *Mouse_move* is another event procedure which controls the current location of the mouse pointer. location of mouse pointer on the object at any time. *Mouse_Down* and *Mouse_move* event procedures of *drawbase* and the explained parameters are used to organize pipe drawing procedure.

Pipes are drawn directly on drawbase. An array, *pipes_array*, holds start and end points of each pipe on the screen. If a pipe consists more then one segment, every vertices of segments are also hold in the same array. The code drawing the lines on *drawbase,* according to coordinates stored in *pipes_array*, is written in *paint event* of drawbase (drawbase is also a picture box).A Pipe can be selected by pressing on any segment constructing the pipe. No other key combination, or command is necessary.

After selecting the menu item "Pressure lines", a variable *pipe_draw* is set to 1. *Mouse_down* event of *drawbase* checks this parameter every time the mouse buttons pressed on it. If *pipe_draw* equals to 1 and button equals to 1 (left button of mouse is indicated by 1) then parameters representing the coordinates X and Y are taken as the coordinates of the starting point of pipe. Other vertices of the line (pipe) are determined by the same way. Every time a new vertex is assigned, using line command, new and previous vertices are connected. When the right mouse button is pressed, the variable *pipe_draw* is set to zero. Number of vertexes for this pipe is recorded to in the same array, and pipe drawing operation is completed.

When the *pipe_draw* parameter is equal to zero, another code written in *Mouse_down* event of drawbase is executed, and searches among the pipes if the X and Y coordinates of the mouse pointer can select a pipe. If a segment of a pipe is close to the picked point, this pipe becomes *selected* and redrawn with the color red. By pressing "Delete" key, the data of the selected pipe is removed from *pipearray*, and lines representing this pipe is cleared from the screen.

By pressing "*shift*" key while selecting pipes, more than one pipe can be selected. If "*shift*" key is not pressed, selecting a pipe cancels the previously selected pipe(s). All of the selected pipes can be deleted at the same time.

Pilot lines are drawn and their vertex coordinates are stored at the same manner. But while drawing the pilot lines on drawbase, *DrawStyle* property of drawbase, which determines the line type of graphics (lines, circles, arcs) drawn on a picturebox, is set to *dashed*.

Another procedure is written for rotation of components, named *rotate*. When a component is selected by pressing left mouse button and "Control" key together, type of the component is assigned to a variable "*selectedelement*", and the number identifying the component is assigned to a variable "*selectedno*". The subroutine *rotate* is called with these two parameters. Every component drawn in picture boxes has a number stored in its array, which represents the current orientation of this component. The number is increased or decreased by 1 depending on direction of rotation. Width and height of the picture box is changed according to its appearance in this new orientation. For example, a directional control valve is drawn in picture box with 90 units width and 30 units height for orientation it is loaded, orientation 1. When it is rotated 90°, width and height values are replaced, width of the picture box becomes 30 units, and height becomes 90 units. The program code placed in paint event which draws the symbol graphic on picture box, controls the orientation number for this element. For every orientation of elements (total 8 orientation for all elements) different parts exist to draw every orientation with different coordinate values. *Resize* procedure explained before, and used for zooming facilities, controls the orientation of each element, and picture boxes are resized accordingly.

Print facility of the program uses "*Printer object*" as the target of graphics methods. *Printer object* is used to control text and graphics printed on paper and to send output directly to the default system printer. *Printer object* is accesses by writing "Printer." as a suffix to the graphics commands. So instead of drawing symbols on picture boxes, when printer object is used as output area, the same schematic is obtained on paper.

Printing subroutine use graphics methods, line, circle, etc., to draw the graphics on paper, and high quality outputs can be obtained depending on the system printer. Another frequently used alternative, *Printform method* offered by Visual Basic is an easy and quick method to print graphics on screen to page, but gives low quality graphics output. So print code for all element's every orientation is written to obtain high quality outputs.

Same code is used for *Print* and *Print Preview* commands. *Print Preview* uses a form named *Preview Form* for output area, and the schematics can be seen on this form as it will be seen on paper, since same code controls the way they are drawn.

To add a new component to the library, extra code is added to draw the symbol without changing any previous code. The group which the new component belongs is determined and its symbol is drawn with *Paintbrush*, a simple painting program comes with Windows. A new picture box is added to the selection form of this group. For example to add a new pressure control valve type, a new picture box is added on pressure control valve selection form. A number is assigned to this new type. On *click event* of this new picture box, *seltype* = (given number to new element) is written. When "Load" key is pressed (which has a standard code for every element type), the array filled for this element with the variable *seltype*. Then, in *paint event* of the picture box which is designed for this component group, the code necesarry to draw the new component type is added. There is an *if..then..else* statement in paint event of the picture box of each group, which checks the element type. Code starting with *elseif type= (given number to new element)* and continues with graphics commands is added as a new alternative in *if..then..else* statement. For every orientation of the new element, code should be written. Width and height of picture box for every orientation of the new element should be written in *relocate* procedure. New code also should be written for printing this new type. Print code for every element group is in module file *maeprint.bas*. Same code written in paint event of the picture box is written in related elements subroutine in this module.

3.5.6 Simulation

For simulation, first of all, a circuit should be constructed by obeying the rules related to the connection of elements as detailed in the previous sections. Then, required data should be entered for pumps, cylinders, pressure control valves, and hydraulic motors.

To calculate velocity of pistons, total flow rate to in the system should be known. Flow rate of pumps can be specified in units of liters per minute (l/min). A form for pump flow rate is reached by locating the cursor on the desired pump and pressing the right mouse button. Data form for pumps is shown in Figure 3.19.

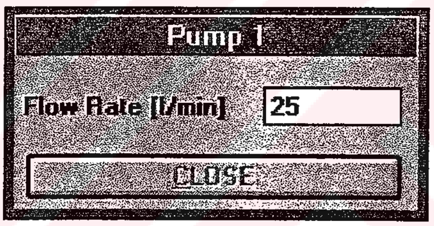

Figure 3.19 Pump data form.

Opening of a variable orifice can be set with a scroll bar placed on a form, without giving a definite value. It is assumed that the pressure drop through the orifice is zero when it is fully opened and flow is blocked when it is closed. On other settings, pressure drop through the orifice is not taken in to consideration. In a circuit, variable orifice is used together with a relief valve to set the flow rate going through it, by sending some portion of pump flow to the tank via the relief valve. In Figure 3.20 the form for setting the opening of variable orifices is shown.

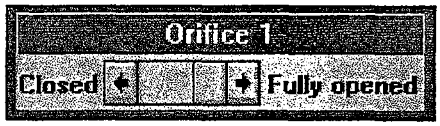

Figure 3.20 Setting opening for variable orifices.

For pistons, by locating the cursor on component and pressing the right mouse button, a form is displayed to enter applied load on piston during extension and retraction in strokes kiloNewton, piston and piston rod diameter in millimeters. Pressing "enter" on each field, sends the focus to the next one. Pressure demand of the piston to overcome the applied load is calculated by the program when all the required values entered. Rod diameter is used in the calculation of back pressure of the piston when another piston is connected to the outlet port of the piston. Data entrance form of cylinders is shown on Figure 3.21 with sample data.

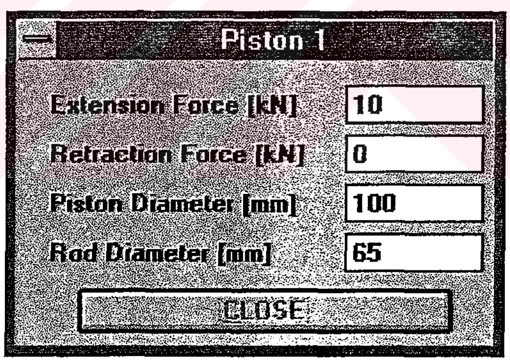

Figure 3.21 Piston data form.

Set pressure of pressure control valves can be entered using a similar form. The form appeares when right mouse button is pressed by locating cursor on the component, and shown in Figure 3.20. The set pressure value is specified in Bars.

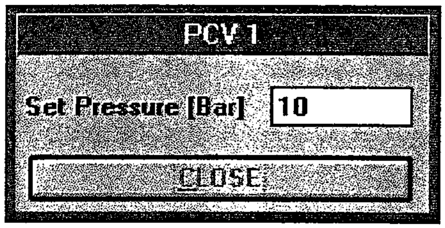

Figure 3.22 Pressure control valve data form.

For relief valves, the pressure value at which the valve just starts to open is called the cracking pressure. The pressure required to handle full pump flow is greater than the cracking pressure. The pressure at full pump flow or the cracking pressure may be specified when referring to the pressure setting of the relief valve. Variable orifices adjust the flow rate passing through them within the pressure range between cracking pressure and full pump flow pressures of the relief valve. On the other hand, DraftSima only uses a single set pressure for relief valves. When the pressure in the system reaches the set value of the relief valve, full pump flow is assumed to be going in to the tank through the relief valve. So speed adjustment with variable orifice is not possible, it can only be used to increase the pressure in the branch and when relief valve setting is reached, to send all the pump flow in to the tank. In Figure 3.23, flow rate versus pressure characteristics for a simple relief valve and the characteristics used by DraftSima are given.

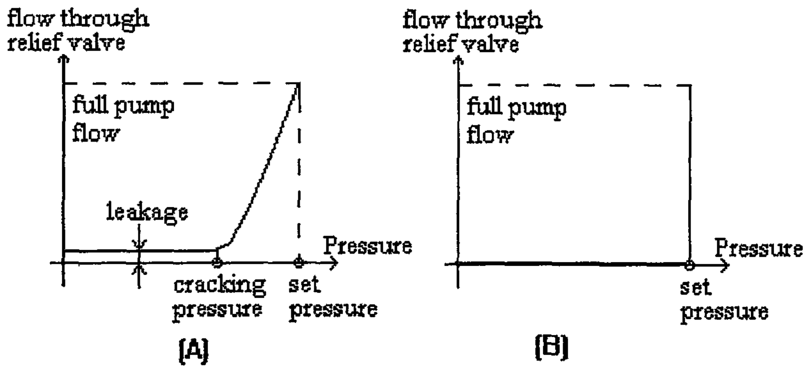

Figure 3.23 A) real case B) The case DraftSima uses.

Simulation can be started by using the items in “Simulation” menu shown in Figure 3.24.

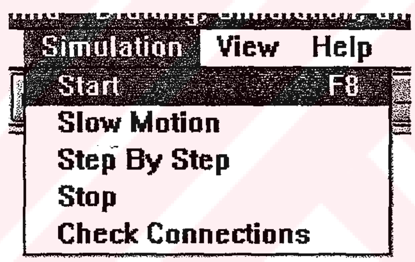

Figure 3.24 Simulation menu.

Before starting the simulation, pipe connections should be checked. Sometimes, endpoints of pipes can be seem to be connected to ports of the elements although they are not. Choosing “Check Connections” item from simulation menu, controls connections of pipes. If any one of the ends of pipe is not connected to a component or a pipe, its color becomes *green*.

To start simulation, "Start" item can be selected from menu, "F8" key can be pressed or from the toolbar, the button with camera picture can be used. The buttons on the toolbar related to simulation are seen in Figure 3.25.

Figure 3.25 Toolbar buttons related to simulation: Start button, Stop button, Next Step button.

During simulation components can not be moved, rotated or deleted. But zoom facilities can be used.

After simulation starts, colors of pipes change to indicate the fluids pressure state. Pressurized lines become red, return lines from actuators become blue and lines which has no flow remains black. Drawing order of the pipes has no importance.

Directional control valve's active spool position can be changed by pressing mouse button when the cursor is on desired spool. Valve shifts to activate the spool. If pilot lines are connected to actuate the directional control valve, active spool is automatically selected and valve shifts automatically.

Stop command from simulation menu stops the simulation. Directional control valves, actuators, and pressure control valves return their original shape and position. Colors of all lines return to black.

If it is desired to observe simulation step by step, "Step by step" menu item is selected. This facility is suitable when explaining the operation of the circuit to someone else, or trying to estimate the next step of operation. After every major movement of components, simulation is suspended and for the next step, program waits for a user input. For example, after shifting the directional control valve, the lines through which pressurized fluid goes turn in to red, but cylinder attached to the end of line does not extend until *next step* button is clicked ,or after cylinder stroke is completed, relief valve is not opened until *next step* button is clicked. *Next step* button is placed on toolbar, with a *footprint picture* on it.

3.5.7 Procedures Applied For Simulation

Simulation code consists of three main parts, determination of connections, determination of flow paths, and animation of the functions of components. The procedures for determination of flow path is called everytime a directional control valve is shifted by the user. When re-determination of fluid flow is necessitated, they are called automatically to restart searching path.

i. *Determination of pipe connections* :

At the drafting stage of the schematics, the order through which the pipes are drawn is not important. A procedure named *connectpipes* is used to fill the array *connections* to recognize the circuit. Procedure takes the X and Y coordinates of each pipe's end points, and searches for the ports of all elements to see if the coordinates match. X and Y coordinates of end points of pipes are taken from *pipearray* which is filled while the pipe is drawn. *Connection* array has 18 columns and 1000 rows. So at most 1000 pipes can be used in the system, but this is not ment to be a limitation of the program. If desired, changing the number 1000 to 10000 in the definition of *connection* array, maximum pipe number can simply be increased to 10000.

For every pipe, components or other pipes connected to each end is written in the corresponding row of *connections* array. Only one element can be connected to each end of the pipes. If an element is connected, no other connection is permitted. At most, three other pipes can be connected at one end of a pipe.

First, second, and third columns of *connections* array contain data related to the component attached to first end of the pipe and fourth, fifth and sixth columns contain the data related to the component connected to second end of the pipe. Instead of a component, if other pipe(s) are connected to the end of the pipe, other columns are used. Columns 7, 8, 9 and 13, 14, 15 are used for pipes connected to first end, and columns 10, 11, 12 and 16, 17, 18 are used for the pipes to the second end of the pipe.

During construction of a circuit, there is no difference between ends of the pipes. User does not deal with which end of pipe is connected to which component. Numbering ends of pipes as first and second is only used in simulation procedures by the program.

Three different items are stored for a component in *connections* array : component type, component no and port of the component. A pipe , first end is connected to a directional control valve end second end is connected to three different pipes and is shown in Figure 3.26. Note that Pipe-1 has one segment and other pipes have two segments. A pipe can consist of more than one segment (at most 20 segments) but all of them are threated as a single pipe. Again, the user never deals with which end of the pipe is connected to which component, drawing order has no importance.

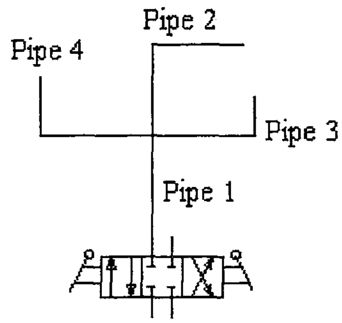

Figure 3.26 A pipe connected to a DCV and three other pipes.

Data stored in to *connections* array for Pipe-1 in figure is given below.

For first end: columns 1, 2, 3 is filled for the connected component

Connections (1,1) = "DCV", component is a directional control valve

Connections (1,2) = "1", component number is 1

Connections (1,3) = "A", port "A" of directional control valve

For second end: columns 4, 5, 6 is filled for the connected component

Connections (1,4) = "Pipe", a pipe is connected to second end

Connections (1,5) = "2", connected pipe's no is 2

Connections (1,6) = "1", connected pipe's first end

For first end: columns 7, 8, 9 is filled for the connected component

Connections (1,7) = "", Empty no other component or pipe at firs end

Connections (1,8) = "", Empty

Connections (1,9) = “”, Empty

For second end: columns 10, 11, 12 is filled for the connected component

Connections (1,10) = “PIPE”, another pipe at second end

Connections (1,11) = “3”, connected pipe’s no is 3

Connections (1,12) = “2”, connected pipe’s second end

For first end: columns 13, 14, 15 is filled for the connected component

Connections (1,13) = “”, Empty: no other component or pipe at first end

Connections (1,14) = “”, Empty

Connections (1,15) = “”, Empty

For second end: columns 16, 17, 18 is filled for the connected component

Connections (1,16) = “PIPE”, another pipe at second end

Connections (1,17) = “4”, connected pipe’s no is 3

Connections (1,18) = “1”, connected pipe’s second end

For Pipe-1 *connections* array is filled as it is given above. Directional control valve is connected to first end of Pipe-1 and pipes 2, 3 and 4 are connected to second end. For all pipes, *connections* array is filled in the same way. Pipe-1 in the figure is also written to row’s of Pipe-2, Pipe-3, Pipe-4 as the connected pipe.

Another array *pilotconnections* are filled for pilot line connections in the same procedure, *connectpipes*.

After *connections* array is filled for all pipes, a procedure named *chkconnections* checks the *connections* array for both ends of every pipe. If nothing is connected at any one of the ends of a pipe, the pipe is drawn in green to warn the user.

ii. Determination of flow path :

After all the components are connected to each other, path of flow from pumps to tank in the circuit is determined. The procedure named *findpath* uses *connections* array to determine the path of fluid flow in the circuit.

A special data type is defined as a *structure* to store elements in branches.

First of all, the pipe is searched in *connections* array which is connected to a pump. If "PUMP" is found to be connected to any one of the ends of a pipe in *connections* array, it is recorded as the first element of the path.

A data structure is created to store different fluid paths, number of components in each path and pressure state of pipes in path. The structure is given below. Fluid from tank can be divided in to branches and goes different parts of the circuit. At every branching point, flow in to different directions are stored as different paths. The data structure is constructed as follows:

Type flowpath, name of the data structure is flowpath

path(1000, 4) As String ,(i%,1)= name of the element

,(i%,2)= no of the element

,(i%,3)= port of the element or end of pipe

,(i%,4)= pressure at this point

pathtotal As Integer ,number of components in this branch

level As String , level of this branch

levelpath(20) As String ,sequence of other branches bringing flow to this branch

branchpressure As Single ,pressure required to do work at end of this branch

End Type , end of type declaration

The data structure named as *flowpath* is defined as an array to use it for every branch. The array *path* used as a part of the structure *flowpath*, is used to store pipes and components in this branch. Integer variable *pathtotal* represents total number of elements in this branch.

The pipe connected to the pump is the first branch and it has a level given by the program as 1. When the flow is first divided in to branches, level of the new branches are given as 2, and new branches divided later are numerated similarly. Level of the branches are stored in the variable *level* in the data structure. *Levelpath* array of the structure stores the numbers of the branches necessary to arrive this branch on the path from the tank. In Figure 3.27 some portion of a circuit is shown to explain what the branch and level is.

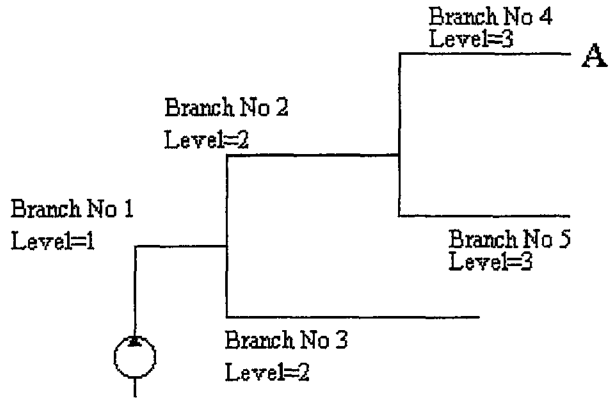

Figure 3.27 A sample of branching.

For the circuit in Figure 3.27 , there are 5 branches. Level of every branch is written in figure. To reach point "A" starting from pump, branches in order 1-2-4 should be followed. In *levelpath* array of data structure reserved for branch-4 , 1-2 (the branches bringing flow from pump to this branch) is written.

After the pipe connected to the pump is found, the element at the other end of the pipe is easily found from the *connections* array. A procedure, named as *redpipe* , is used for drawing the pipe in red to show that this is a pressurized line. Another procedure is called to determine the state of the component connected other end of the pipe.

According to the type of component on the other end of the pipe, different situations may occur:

a) ***If the component is another pipe***: Algorithm checks if there is more than one pipe connected to this point. If more than one pipe exists, program recognizes that the branch is divided in to new branches, and finishes writing the data for this branch. Number of branches increased by the number equal to total number of pipes connected to this pipe. And they are followed as new branches. On the other hand, if there is a single pipe connected to the end of previous pipe, procedure continues on searching.

b) ***If the component is a directional control valve***: According to the activated spool position and entrance port of the valve, different situations may occur. If the inflow is given to another port, the pipe at the outlet port of the directional control valve is found, and procedure continues on searching the elements on branch. Bu if the port which the pipe is connected is blocked, i.e. fluid flow is stopped, than a value as "BLOCKED" is given to the variable *branchpressure* of this branch and operation is stopped.

c) ***if the component is a tank***: Zero is given as the *branchpressure* and searching operation is stopped.

d) ***if the component is a pressure control valve*** : If the valve's state is closed, and this is not a pilot operated valve, than set pressure of the valve is given as the *branchpressure* and searching operation is stopped. If the valve's state is open (opened with the signal of a pilot line before), the pipe connected to other port of the valve is found, and searching operation continues.

e) ***if the component is a cylinder***: Searching the branch is stopped when a cylinder is reached. If the cylinder is at the position which the inflow cannot move it anymore (if it is at the end of its stroke), "BLOCKED" is assigned to branchpressure. On the other hand, according to the port the fluid enters, pressure value obtained by using extension force or retraction force, piston diameter and piston rod diameter, is assigned to branchpressure and searching operation is stopped.

f) ***if the component is a flow control valve***: If the component gives permission to fluid flow entering from this port, the pipe connected to other port is found and searching operation continues. If the component does not permit fluid flow in this direction, operation is ended and "BLOCKED" is given as the *branchpressure*.

g) ***if the component is an accessory***: For filter, flowmeter, heater and cooler, flow is directly given to the other port of the component. Pressure gages and accumulators are not taken as standard components.

After all branches are determined, *findpath* procedure stops.

iii. Determination of components to simulate

The rule of thumb for fluid flow is fluid goes in to the easiest path in the circuit. That means, components in the branch with minimum pressure value is activated first.

A procedure named *simulate* checks pressure values assigned to branches, and the branch or branches having the minimum pressure value is found. Branches called *internal branches* are not taken in to consideration. *Internal branches* are the branches which do not arrive actuators or pressure control valves, only used to divide flow in to sub branches. Branches 1, 2,3 in Figure 3.27 are *internal branches.*

After finding the branch or branches with minimum pressure values, component at the end of the branch is examined. According to type of the component, one of the following cases considered.

Case 1: If the component at the end of the branch is a *cylinder*, a procedure, "*movepistons*", is called and motion of the cylinder is animated. After the piston reaches end of its stroke, *branchpressure* of this branch is set to "BLOCKED", and the branch with the next minimum of *brancpressure* is found for animating.

Case 2: If the component at the end of the branch is a relief valve, its graphical image is redrawn in opened position, and procedure is ended.

Case 3 : If the component at the end of the branch is a sequence valve, (the valve permits fluid flow when the pressure in front of it reaches its set value), the valve's graphical image is redrawn in opened position and a subroutine named as *continue_on_branch* is called to find components connected to the path continuing. When another component is found on this branch, *branchpressure* is re-determined with the value obtained from this component and it is checked whether or not the value is smaller than the minimum of *branchpressures* found before.

Case 4 : If the component at the end of the branch is tank, that means *branchpressure* is zero: no resistance to flow, procedure is ended.

Only the components explained above can change situation with the pressure applied. Any other component either permits or blocks fluid flow without dealing whatever the pressure is.

3.5.8 Pilot Lines

Pilot lines are used to control some of the components with the pressure state of connected point of the circuit. A procedure called *sendsignal* is used to determine the connections of pilot lines and control pressure controlled components.

After *findpath* procedure determines the pipes with pressurized fluid, *sendsignal* procedure checks pilot line connections. The pressure value where the pilot line takes signal is set as the pressure value of this pilot line. The component is activated by checking this value. Directional control valve with pressure operated actuators, pilot operated pressure control valves and pilot operated check valve can be controlled with pilot lines.

If the component to control is a directional control valve, value of pressure has no importance. Directional control valve is shifted to the spool at the side of pilot line connections. The pilot line is signed as "*done_work*" and for the new position of valve, flow in the system is re-controlled. During this control, if *sendsignal* procedure is called again when a pilot line is considered, the variable in the array pilotlines is controlled for the pilot line in consideration to determine whether this pilot line is the one which caused this re-control process. If this is the pilot line which is just send the signal, it is omitted to prevent the program from going in to a infinite loop.

If the component which receives signal is a pilot operated pressure control valve, pressure value of the pilot line is compared with the set value of the valve. If it is greater, the corresponding field of *pcvarray* for this valve is set to "open". Then the procedure *continue_on_branch* is called to find the next part of the branch starting from the exit of this pressure control valve. If this is a pilot operated relief valve, fluid is send to tank and no extra work is done.

If the component is pilot operated check valve, it is set to permit flow in both directions. Again *continue_on_branch* is called to find the next part of the branch starting from the exit of this component.

3.5.9 Accumulators.

Accumulators are assumed to be filled after a cylinder on the branch is completed its stroke, and before the relief valve is opened. After every movement of cylinders or directional control valves, accumulators connected to the circuit are controlled. If there is no flow in to an actuator on the branch or no free flow towards the tank, accumulator is filled. No pressure values for accumulators. Accumulators are treated as pumps.

For an accumulator filled before, if there is a path to reach an actuator, and the path from accumulator to pump is closed, accumulators feeds the system as a pump. Again it is assumed that, the amount of fluid stored in accumulator is enough to feed all the actuators in the system.

3.5.10 Pressure Gages

Pressure gages are not taken in to account while connecting pipes. A procedure named *gage* checks only the coordinates of pressure gages and finds the branches they are connected. Then with some basic principles, pressure value at the point the pressure gage is attached is determined.

Rules used to determine the pressure at the desired point are:

1. If a cylinder is actuating, pressure value at every line carrying pressurized fluid is equal to the pressure required to actuate this cylinder.

2. If there is an open relief valve, pressure value at every line carrying pressurized fluid is equal to the set pressure of this relief valve.

3. If there is an open sequence valve, and a piston moving in the system, pressure value at every line carrying pressurized fluid is equal to the greater value.

CHAPTER 4

CASE STUDIES

In this chapter, hydraulic power circuit schematics that are constructed by DraftSima and simulation stages of these circuits will be given in detail. Case studies are chosen among the classical hydraulic power circuits that are explained in detail in Chapter 2. DraftSima can simulate all the typical circuits given in Section 2.6.

In figures of this chapter, red lines represents the pipes carrying pressurized fluid, blue lines represents the return-to-tank lines, and black lines represents the pipes in which there is no fluid flow.

4.1 A Single Acting Cylinder Circuit

The circuit given in Figure 2.14 is constructed in DraftSima by selecting a single acting cylinder from actuators library, a pump from pumps library, a two position three way directional control valve from directional control valve library, a relief valve from pressure control valve library, a filter from accessories library and two tanks. Components are connected with pipes.

In Figure 4.1, schematic of this circuit printed by print utility of DraftSima is given.

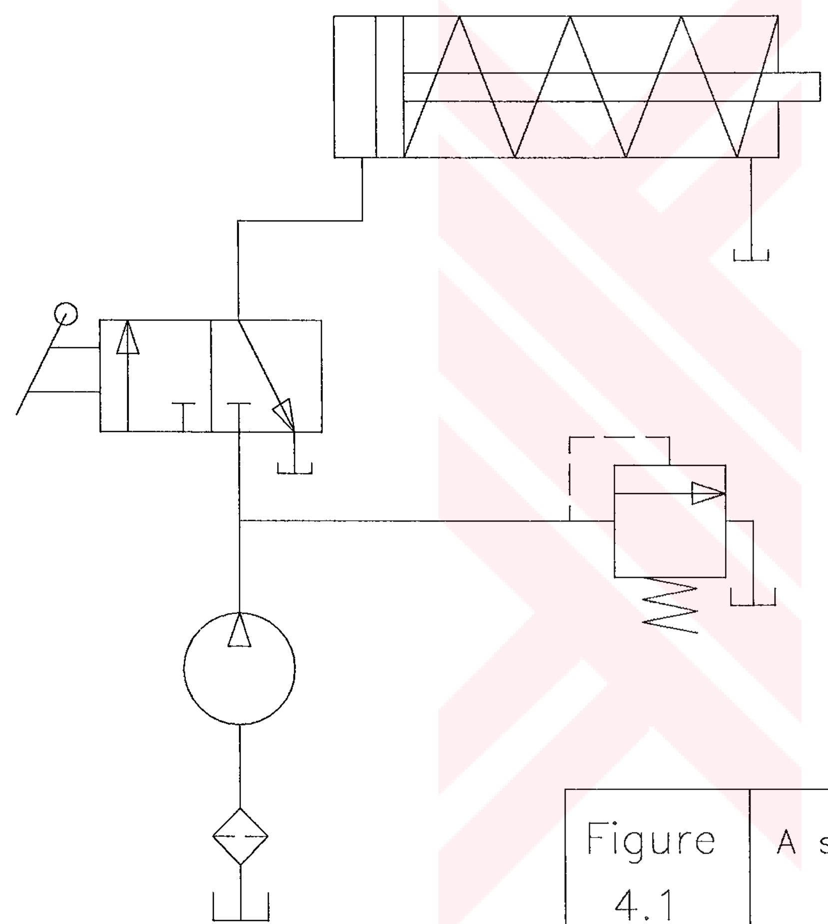

Figure 4.1 A single acting, spring returned cylinder control circuit

When "Simulate" item is clicked from "Simulation" menu, pump is started and the pressurized lines are shown in the color red. Figure 4.2 shows the view of the screen when simulation is started. For the position of directional control valve, fluid flow is not send to cylinder, pressure increases in the system and relief valve is opened. Pump flow is directed to tank via the relief valve. When the directional control valve is shifted to other position, pump flow is directed to cylinder and it extends. In Figure 4.3 the screen during cylinder extension is seen. After the cylinder ends its stroke, the system pressure again rises and when it reaches the set pressure value of the relief valve, pump flow is directed to the tank again via the relief valve as it is seen in Figure 4.4. In Figure 4.5 next step of operation is seen. When the directional control valve is shifted to its initial position, pump flow is directed to tank via the relief valve and spring pushes the piston to its initial position. Fluid stored in piston is send to the tank.

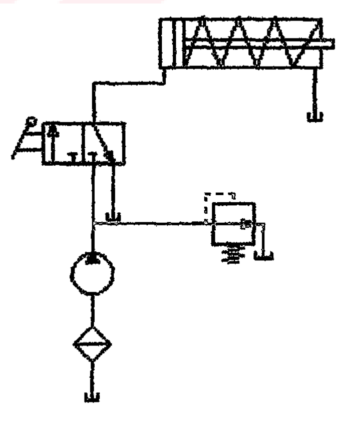

Figure 4.2 Fluid flow is directed to tank via the relief valve

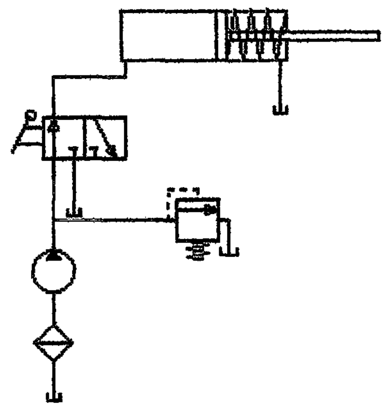

Figure 4.3 Pump flow is directed to cylinder, and it extends.

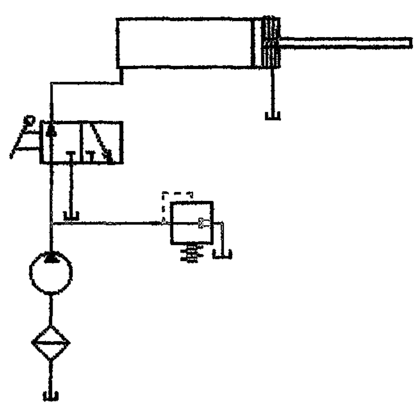

Figure 4.4 When the piston reaches end position, relief valve is opened

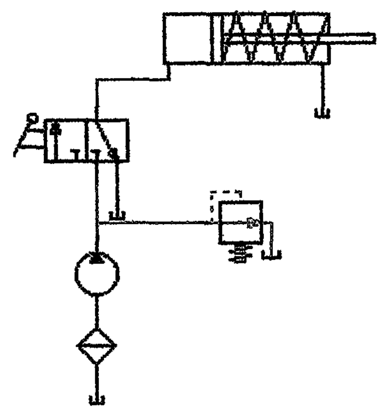

Figure 4.5 Retraction of the cylinder. Pump flow is directed to tank via the relief valve

4.2 Two Cylinders Connected in Parallel

The circuit explained in Section 2.6.3 and shown in Figure 2.17 is drafted with DraftSima. For both pistons same dimensions of cylinder and rod diameters are given. Extension force on the cylinder labeled as "cylinder 1" in Figure 4.6 is given as 10 kN, and the load on the cylinder labeled as "cylinder 2" is given as 15 kN. Relief valve is set to 150 Bar. When the pump is started, (i.e. simulation starts), pump flow is send directly to tank at atmospheric pressure by center position of directional control valve as shown in Figure 4.6. Since the easiest path to the tank is used by fluid, pressure in the system does not increase, so relief valve is not opened. When the directional control valve is shifted to direct the flow in to the cylinders, first the cylinder with smaller load on it extends (Figure 4.7) . After it ends its stroke, pressure rises and second cylinder is pushed.(Figure 4.8). After the extension of cylinder 2 is completed, relief valve is opened and flow is directed to tank (Figure 4.9). Third position of directional control valve is used to retract the cylinders. Since there are no retraction force applied, both cylinders retract together as it is seen in Figure 4.10. After retraction is completed, pump flow is directed to tank through the relief valve as shown in Figure 4.11. In all figures, pressurized lines are drawn in red, return lines are drawn in blue, and the line which have no fluid flow is drawn in black.

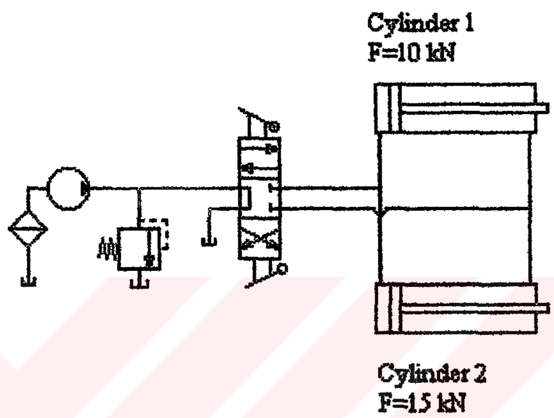

Figure 4.6 When the system started.

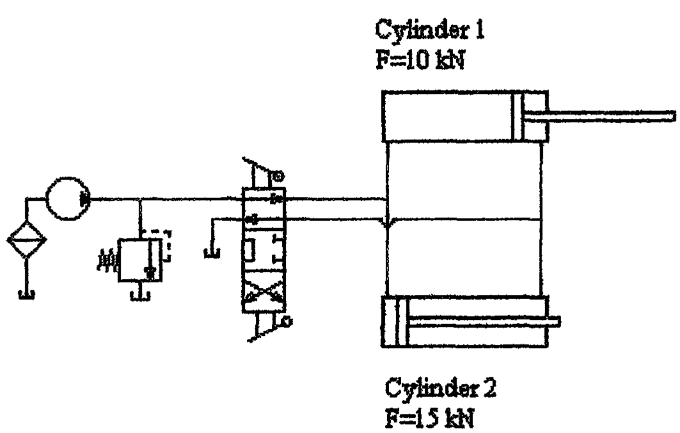

Figure 4.7 Cylinder with smaller force on it, extends first

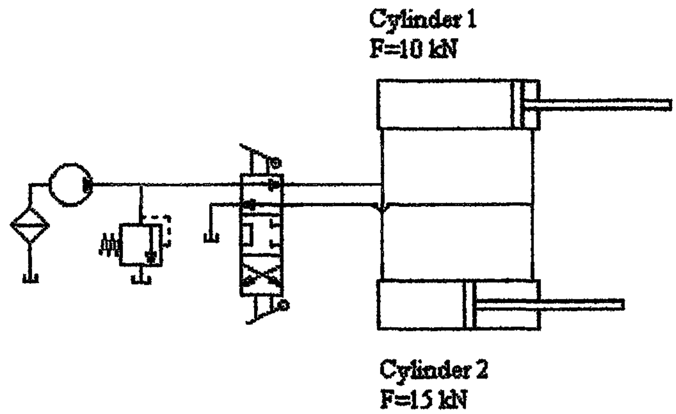

Figure 4.8 Then the cylinder with larger force on it extends.

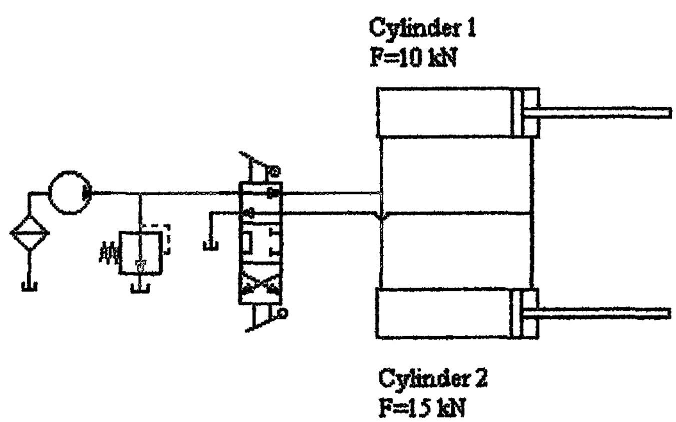

Figure 4.9 Relief valve is opened after both cylinders complete their extension stroke

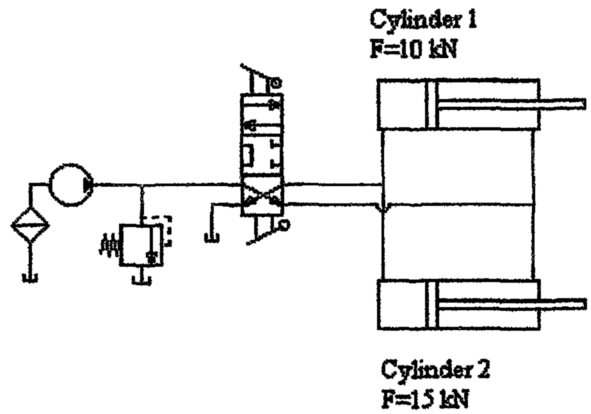

Figure 4.10 For the position of DCV, cylinders retract together

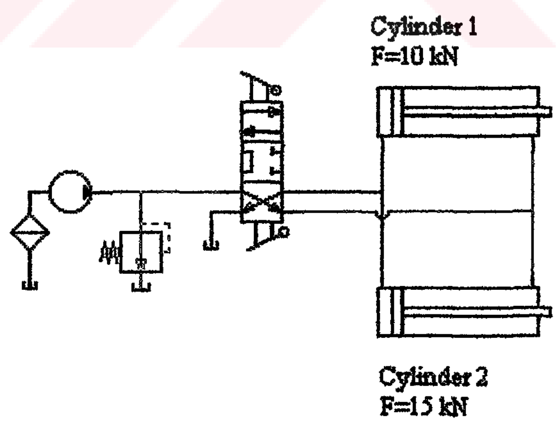

Figure 4.11 After retraction is completed, relief valve is opened

4.3 Cylinder Sequencing Circuit

The cylinder sequencing circuit explained in Section 2.6.4 and shown in Figure 2.17 is drafted with DraftSima. For both cylinders, extension forces are set as 1 kN, piston diameters are set as 100 mm and rod diameters are set as 65 mm. Opening pressure values for sequence valve 1 (PCV 1 in Figure 4.12) is set as 10 Bar and for sequence valve 2 (PCV 2 in Figure 4.12) is set as 15 Bar. Relief valve is set to 20 Bar. When the pump is started, for the position of directional control valve shown in Figure 4.12, relief valve is opened and pump flow is directed to the tank. When the directional control valve is shifted to the position shown in Figure 4.13, fluid flow reaches to both of the sequence valves. The sequence valve which is set to 10 bar is opened first, and the cylinder connected to this branch extends (Figure 4.14) . After its stroke is completed, pressure of the system increases and the sequence valve which is set to 15 Bar is opened. As it is seen in Figure 4.15, the cylinder connected in to this branch is extended. After its stroke is completed, pressure increases and relief valve is opened (Figure 4.16). When the directional control valve is shifted to other position, both cylinders retract together. Fluid coming from pistons, goes to tank through the check valves connected to sequence valves as shown in Figure 4.17

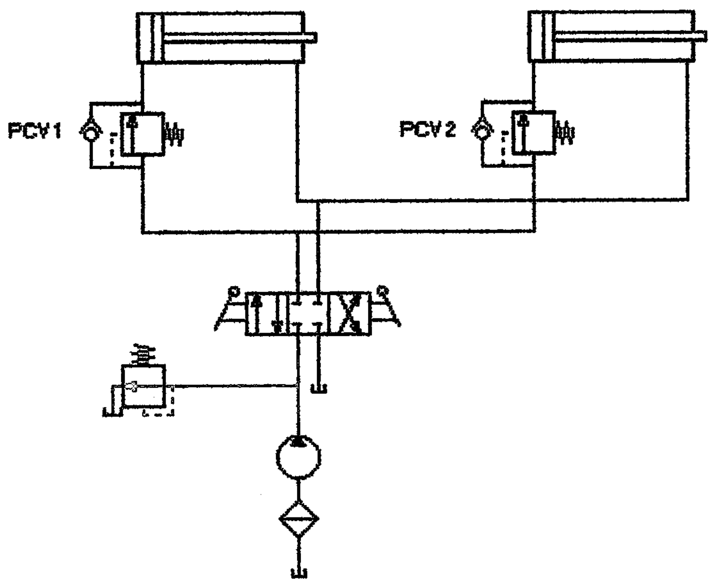

Figure 4.12 When the cylinder sequencing circuit is just started

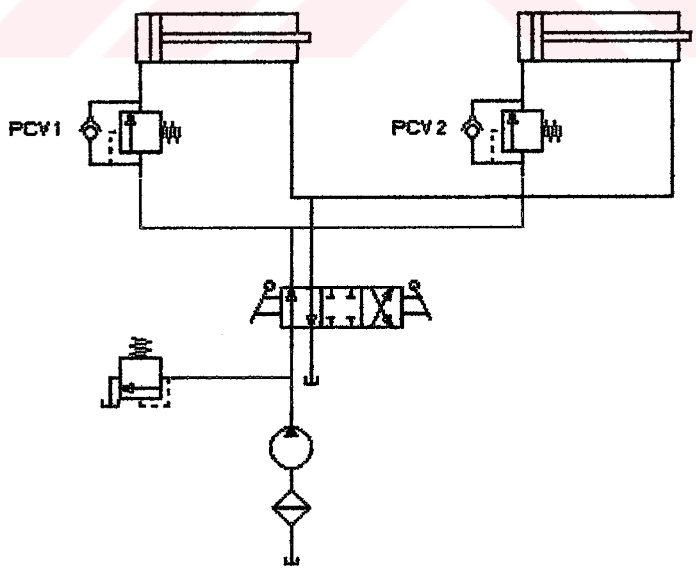

Figure 4.13 Flow reached to sequence valves

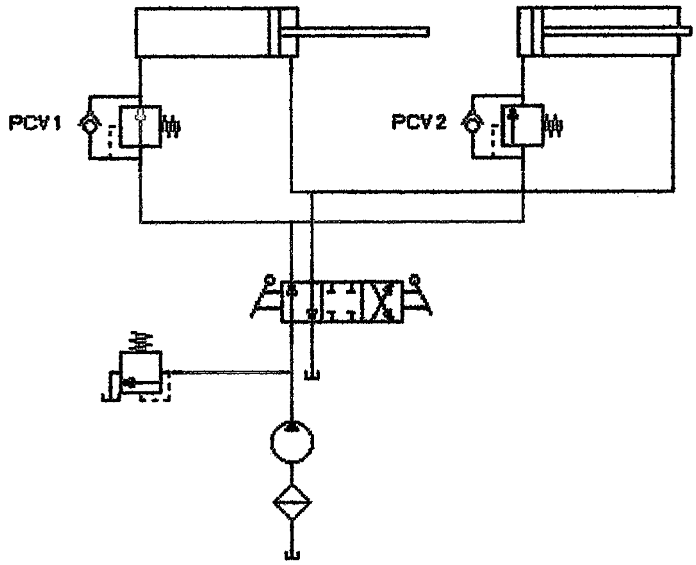

Figure 4.14 Sequence valve with smaller setting is opened

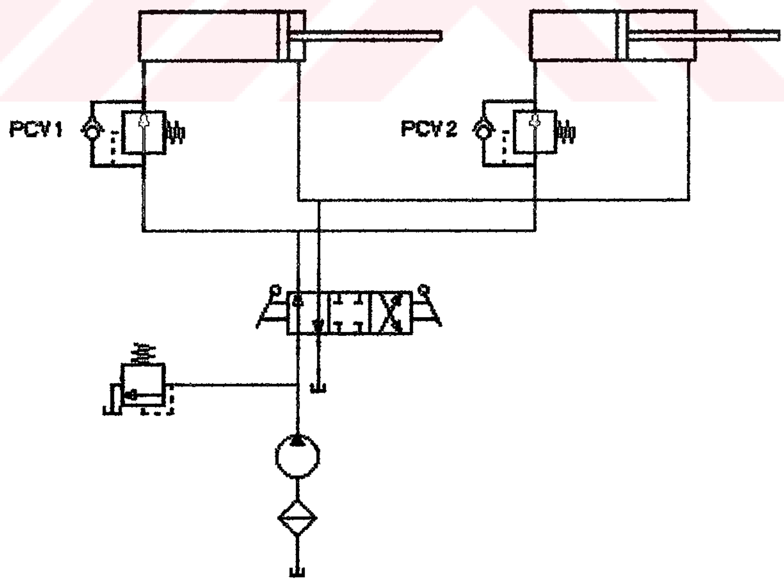

Figure 4.15 Sequence valve with larger setting is opened

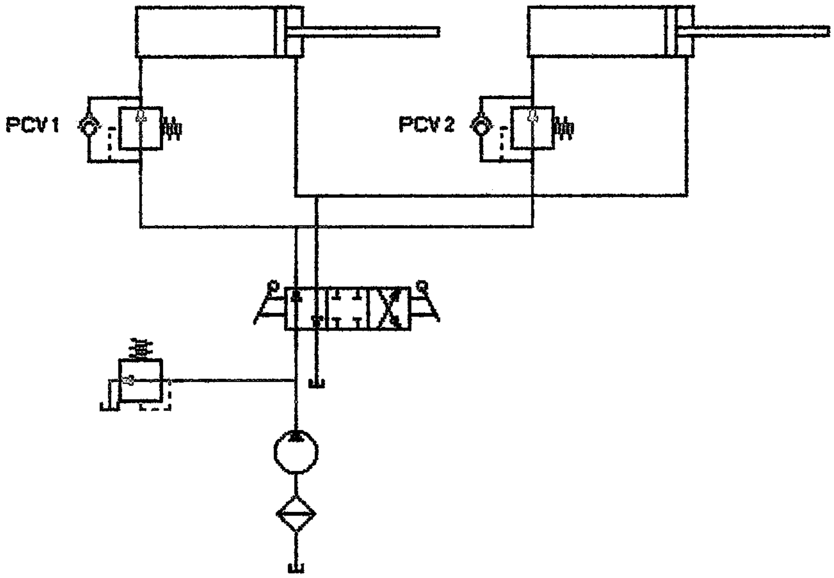

Figure 4.16 Relief valve is opened after both cylinders complete their strokes

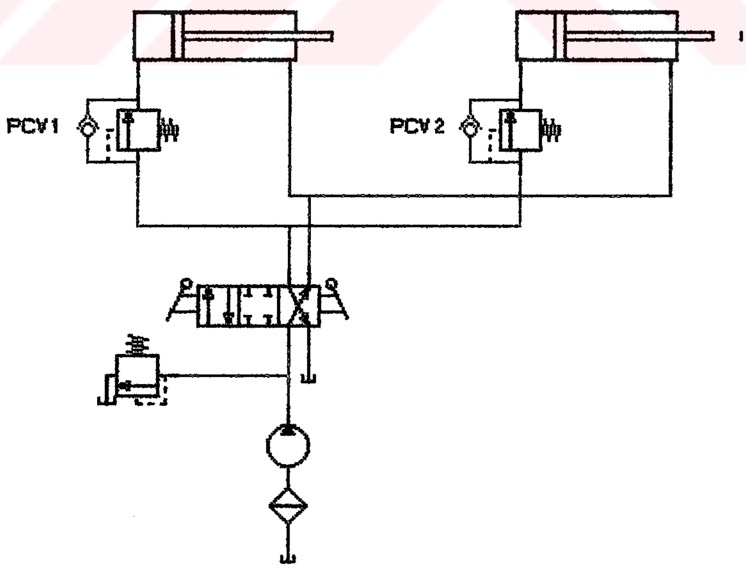

Figure 4.17 Both cylinders retract together

4.4 Continuous Cylinder Reciprocating System

The circuit explained in Section 2.6.5 and shown in Figure 2.20 is drafted with DraftSima. For the position of directional control valve shown in Figure 4.18, pump flow is directed to cylinder, and it is extended. After the extension stroke is completed, pressure in the line increases and the sequence valve 1 (PCV 1) is opened. The pilot line connected to PCV 1 sends a signal to directional control valve. In Figure 4.19, the moment just the pilot line is pressurized, is shown. The pilot line gives the signal to the directional control valve and shifts its position, and the cylinder starts to retract (Figure 4.20). After retraction is completed, PCV 2 is opened and the pilot line gives a signal to the directional control valve to shift its position, and the cylinder extends again. In Figure 4.21 directional control valve is shown to be returned its initial position. The cylinder reciprocates continuously until the pump is stopped

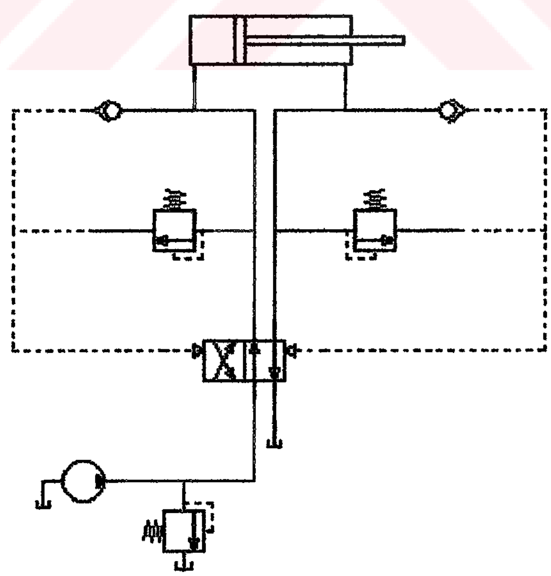

Figure 4.18 First the cylinder extends.

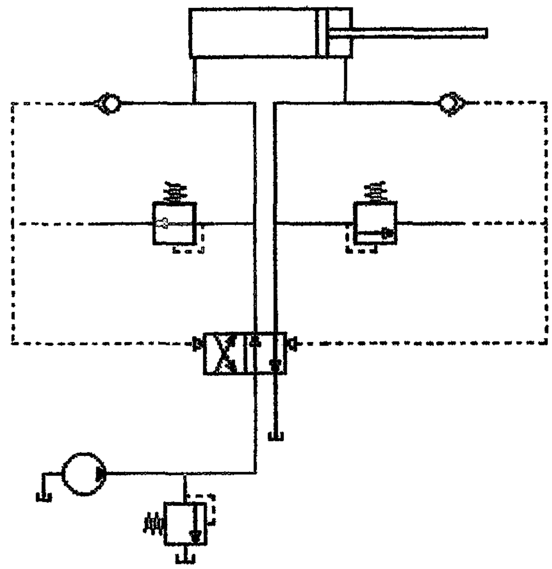

Figure 4.19 After the cylinder ends its stroke, PCV 1 is opened and a signal is send to directional control valve.

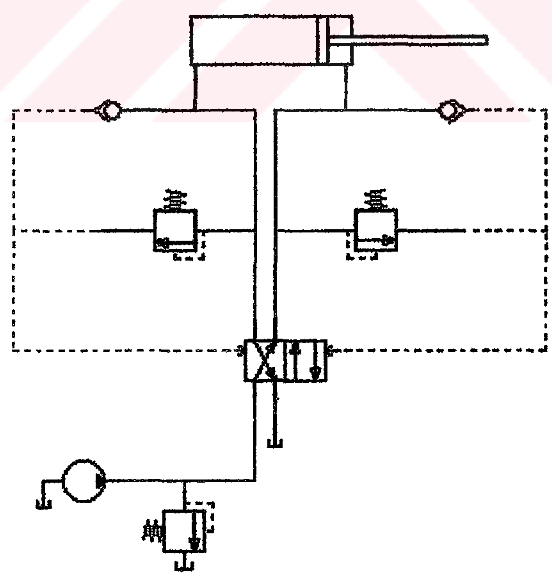

Figure 4.20 Directional control valve is shifted and the cylinder retracts

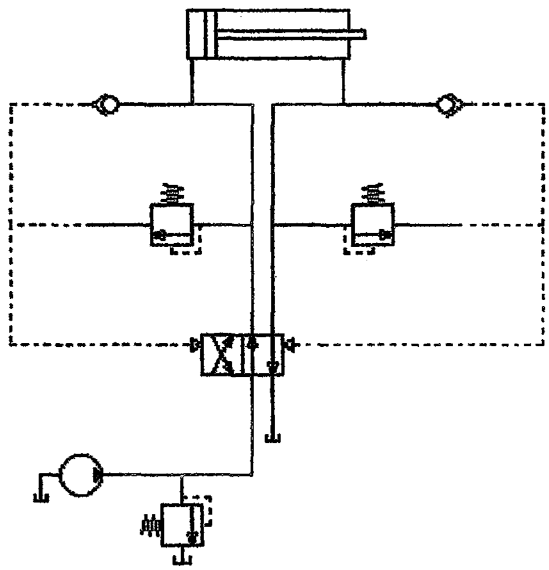

Figure 4.21 After the retraction is completed, PCV 2 is opened and the signal shifts the directional control valve to initial position

CHAPTER 5

CONCLUSION

A software, DraftSima, is developed for drafting and simulation of hydraulic power circuits with some limitations. Software is written in Visual Basic v3.0 Professional Edition, and runs under Microsoft Windows operating environment. Source code of the program contains approximately 30.000 lines.

Standard hydraulic component symbols are taken from ISO 1219-1 (TS 1306), so most of the circuits containing standard hydraulic components can be constructed using DraftSima. Instead of drawing a component symbol in full details manually, choosing it from the library of DraftSima saves important time of the designer. Circuits constructed in the screen can be printed on paper. So software can be used efficiently in design stages of a hydraulic power circuit.

Documents containing hydraulic circuit schematics can be easily prepared with the help of DraftSima. Schematics can be drafted in DraftSima, and using copy-paste facility of the operating environment, can be taken in a document prepared in any word processing software. This is a practical way of preparing reports or course notes which includes hydraulic power circuits. Figures in this thesis document is drafted in DraftSima and taken in to document by using clipboard of Windows.

Steady state operation of an open-loop hydraulic power circuit can be simulated by animating the functions of components. Circuits found in text books or course notes related to hydraulics, can be simulated by using DraftSima. So software can be successfully used as an educational package. To explain operation of complex circuits, simulation can be observed step by step which makes understanding the operation easier.

Developing a software for general case, which can simulate any kind of circuits with no limitations is not the scope of this thesis. But the developed software can be used for most of the circuits that are used for practical applications. As a future work, simulation of closed-loop circuits can be added to DraftSima.

Symbol library of the program can not be extended by the user without modifying the source code. But program is written in a modular form to enable adding new components in to library without changing the existing code, but adding new code for the drawing and simulating the new component. A utility can be added to DraftSima in future, for adding new components in to library at run time of the program, without modifying the source code.

REFERENCES

[1] ISO 1219-1 FLUID POWER SYSTEMS AND COMPONENTS, (1991)

[2] TS 1360 FLUID POWER SYMBOLS, (1975)

[3] HYDRAUSIM v4.03, Copyright by Famic Inc, 1993.

[4] HYDROWORKS, Copyright by TechTeam Inc, 1992

[5] HYDRAMOTION Release 6, Cd-Systems & Eshed Robotec ltd, 1993

[6] ME 481 Industrial Fluid Power, Course Notes by Y. Samim ÜNLÜSOY, Mechanical Engineering Department, Middle East Technical University, 1995

[7] Esposito, A., FLUID POWER WITH APPLICATIONS, Prentice Hall, London (1994).

[8] Henke, R.W., FLUID POWER SYSTEMS AND CIRCUITS, Penton Publishing Inc., Ohio (1983).

[9] Wolansky, D.William, Nagohosian, J., Henke, R.W., FUNDAMENTALS OF FLUID POWER, Mifflin, Boston (1977).

[10] Pinches, M.J., Ashby, J.G., , POWER HYDRAULICS, Prentice-Hall, New York (1989).

[11] Sullivian, J. A., FLUID POWER : THEORY AND APPLICATIONS, Reston Publishing Company, Virginia (1982).