THE DEVELOPMENT OF SOFTWARE TECHNIQUES FOR
DIGITAL SIMULATION OF FLUID POWER SYSTEMS

Seung Ho Cho*  Kee Hyun Shin**  Kyo Il Lee***

* Department of Mechanical Engineering,
  Hongik University, Seoul 121-791, Korea
** Department of Mechanical Design and Production Engineering,
  Konkuk University, Seoul 133-701, Korea
*** School of Mechanical and Aerospace Engineering,
  Seoul National University, Seoul 151-742, Korea

Abstract

This paper describes a computer software package developed to assist the designer of fluid power systems. A variety of component models are selected from the icon library and system circuit can be sketched on the graphic monitor. The input data for component can be provided from the theoretical and experimental analysis, and from manufacture's catalog. For each component the modularized modeling and programming techniques are presented, and for each system the node table is automatically generated to check the continuity and compatibility conditions. A vector icon editor is developed under Motif and imbedded to facilitate the addition of new components. Being the X-window system based on Unix, this software has merits in portability, extensibility, and multi-user environment. The simulation results are confirmed to be consistent with those of other software and experiments.

Keywords

Compatibility, Port, Node, Icon, Circuit

1 Introduction

The fluid power systems are widely utilized in industry including construction equipments, automobiles, airplanes, and machine tools. The fluid power systems usually work as actuators or controllers in the system and determine the performance of the overall system. In the design of modern fluid power systems, sensors, controllers, and other electrical/electronic and mechanical devices are used together, which requires us broaden engineering knowledge to meet the design specification of fluid power systems. Moreover, as the number of the hydraulic components in the system increases, it becomes more difficult to anticipate the performance of the system due to the inherent nonlinearities in the hydraulic components. The computer simulation technique can be an effective tool to attack this kind of situations in the analysis and design of fluid power systems. Concerning fluid power control, there exist several software packages over the world. [1,2,3,4] This paper will present a computer program which will help an engineer to analize and design fluid power systems. The program is to be designed such that the following performances are achieved. A fluid power system can graphically be configured using standard symbols. The necessary data for each component can be easily entered through a graphic user interface.
interface. The dynamic characteristics of the system can be repeatedly simulated in time and frequency domain. The simulation results can be displayed in a table or plot format on the screen and finally the results can be printed out once the simulation is through. This program is a one year product and it is now under development to the effect that all the fluid power programs developed during the fifteen years in Korea would be integrated to this program.

2 Structure of Software

The program consists of four modules: system configuration, system check, system analysis and design, and data visualization modules.

2.1 System Configuration, Check

The user can graphically configure a fluid power system using standard hydraulic symbols in the component library. The configuration, editing, and reconfiguration of a hydraulic system can be easily done by using the graphic editing function installed in the program. Whenever a user selects a component to configure a fluid power system, it is required to provide the necessary data for the analysis. The component data is able to be built through manufacturer’s catalog, experiments, and analysis. System components are classified into four classes: hydraulic, pneumatic, mechanical, sensor and controller components. Once a system is configured by the user, the system should be checked whether all the components are connected in a compatible way before the system is analyzed. The program automatically tests the compatibility and continuity between the connected components in the configured system and reports the results to the user.

2.2 System Analysis and Design, Data Visualization

Once the compatibility of the configured system is proved, the user may want to proceed to the analysis and design stage of the program. In the analysis and design stage, the required data should be provided to the program before the simulation is carried out. The program provides the functions of static and dynamic analysis of the system, and static analysis usually proceeds the dynamic analysis. The data visualization module presents the simulation results to the user in a graphic form. It provides the user with several convenient functions: various form of data set, cartesian x-y plot, bit-map and vector fonts, zoom, postscript, etc..

3 Programming Principles of Fluid Power System

In the design of program a unified approach is applied to the fluid power system. In view of power transfer between components, the causality is realized through effort variable and flow variable. The fluid power system can also be described in terms of across variable and through variable, which facilitate the test of compatibility and continuity condition in nodes or junctions of the system.

3.1 System Variables, Causality

Engineering systems are usually composed of mechanical, electrical/electronic, hydraulic components or the combination of these components. The dynamic characteristics of a system is determined by energy storage, energy consumption, energy source components, and load. A system interacts with external circumstances through the input and output variables. The variables of a system (e.g., force(F) and velocity(V) of a mechanical system, the current(I) and voltage(E) of an electrical system etc.) can be classified into T-type variable (through variable) and A-type variable (across variable). The energy storage component can be classified into T-type and A-type components according to its characteristics of energy storage principle. A port is defined as a place in a component to which other component can be connected. Power is transferred through port between components. In general, a physical component has more than one port. Each port has A-type and T-type variables and one of these variables becomes input and the other is output. On the other hand, the input and output variables are represented as the causal stroke in the bond graph.

3.2 Compatibility and Continuity, Data Structure for Simulation

The following two conditions should be met at each node in each moment during the simulation of dynamic characteristics of mechanical, electrical, or hydraulic systems in the time domain.

- compatibility condition of across variable at the node
- continuity condition of through variable at the node

For example, the pressure at a node in a hydraulic system should be same regardless of the amount of the fluid flow from each port to the node (compatibility condition of across variable). And, for each
node, the sum of fluid flow into a node should be zero. Otherwise, the pressure at a node will be changed due to the unbalance of fluid flow (continuity condition of through variable). Each system component has the following data:

- number of ports
- number of state variables

For each port:
- port type
- port input/output type
- address of the port input
- port output
- connected node number

For each state variable:
- state type
- state

In general, as the number of ports and state variables of a system component is different and are usually more than two, the necessary memory was allocated after the component was identified. Thus, the type for ports and state variables (port type, input/output type, output, state type, state) was declared as pointer. Since the input for each port is decided at the node, it is declared as the pointer of the pointer which has the address of the node variable.

4 User Interface Program

In this paper it is intended that the user can design hydraulic circuit using graphic user interface. The program is realized on Motif of X-Window, and it is named Idea.fp (Integral design engineering assistant.fluid power).

4.1 Structure of Idea.fp

4.1.1 Idea.fp MainWindow

The Idea.fp MainWindow is based on Motif of X Window. The MainWindow is composed of five main menus (File, Edit, Check, Simulation, Presentation), list widget which represents component list, and under which there exists six buttons (View, Edit, Line, Place, Refresh, Comp Data). Beside these buttons there are scale widget, which can magnify or reduce the hydraulic circuit. The MainWindow of Idea.fp is shown in Fig.1.

4.1.2 Overall Data Structure of Idea.fp

In order that the member function can manipulate the contents of the data structure, the address of data structure is necessary. Therefore the first

![Fig.1 MainWindow of Idea.fp](image-url)
parameter of member function corresponds to the address of data structure. At the top of overall data structure there exists HyMainStruct, which acts as parents of all data structure. The data structure HyMainScreen manages the circuit design window. Moreover as a data structure concerning hydraulic circuit, there exist the LsSDObject which expresses hydraulic component in DrawingArea, the HyLine which expresses line, and the LsSDStruct which manages the list of LsSDObject and Hyline. As a data structure used in the process of designing hydraulic circuit, there exist HyCompPlacing which is useful for designing component in DrawingArea and HyRoutingObject which is useful for designing line connecting ports between component.

4.2 Function of Idea.fp

4.2.1 Hydraulic Circuit Design, DrawingArea, and MouseEvent

Hydraulic circuit design is done in DrawingArea widget, which is called simply DrawingArea. The design process in DrawingArea consists of four kinds, i.e., hydraulic component design, line design, correction of hydraulic circuit, and hydraulic component copy from a Clipboard. When all MouseEvents are transferred, the present state of DrawingArea is examined and call for the corresponding function. The data structure for DrawingArea is composed of HyDrawingContext and HydrawingArea. The data for DrawingArea is divided into three kinds of data, i.e., data for hydraulic component design, data for storing hydraulic circuits, and data for drawing figures in DrawingArea.

4.2.2 Component Design and Data Structure of Component

The component design can be processed if only the component is on graphic monitor at DrawingArea. For this if the CompPlace button is on, the design state starts, and all MouseEvents are transferred to the component design procedure. The component informations described on DrawingArea is stored at LsSDObject. Also the data structure for component information, which is called HyCompInfo, is necessary. The data structure for hydraulic component is composed of HylconClass, HyCompInfo, and HyPortInfo, which respectively stores information for hydraulic component class, hydraulic component itself, and hydraulic component port.

4.2.3 Line Design and Data Structure of Hydraulic Circuit

Designing a line follows the same process as designing a component. If the state of DrawingArea is at the state of line design in HyDrawingAreaEvent, all MouseEvents are transferred to the line design function, the name of which is HyLineObjectProc. The hydraulic circuit designed by user consists of a couple of informations, that is, the information for component and the information for line, which are stored in LsSDObject structure and in HyLine respectively.

4.2.4 Parameter Input, Class and Icon

In order to receive particular input from user, various dialog boxes are provided in Idea.fp. In order to manage dialog box a data structure is assigned to a each dialog box, which ends with Dlg. Parameter input method of component is through a dialog box, whose name is component data sheet and the data structure to produce and to manage it is HyCompSheetDlg.and the related source files are contained in lhcmpsht.c. Each component has its own information and it is contained in HyCompInfo.

4.3 Formulation of Node Table

In order to simulate the designed hydraulic circuit, it is necessary to formulate a node table. In Idea.fp the hydraulic circuit is converted into graph, from which node table is constructed. A procedure of formulating a virtual graph from a hydraulic circuit is shown in Fig.2. That is, the graph shown in Fig.2 represents a graph data structure to formulate a node table. The transformed graph is composed of three graph data structures. They respectively correspond to the graph that the number 1 port of SH4111 becomes a number 1 vertex, the graph that the number 3 port of SH9100 becomes a number 5 vertex, and the graph that the number 3 port of SH1110 becomes a number 10 vertex. Starting from a port of a component and going along the line, the crossing point becomes a vertex. Then the ports of component are converted into vertex. All ports connected by a transformed graph have the same node number. Assigning node number to a port is based on a graph. Visiting a vertex, if it is a port, it can receive a node number. Formulating a node table from a hydraulic circuit becomes a problem of deriving a graph and visiting it. To construct a graph data structure the number 1 port of an arbitrary component of circuit is set to a vertex, the line connected to it is considered to be an arc. The
Fig. 2 Graph Data Structure for Node Table

Fig. 3 MainWindow of IconEditor
point where several points encounter is considered to be a vertex, all lines connected from it are thought to be an arc. Converting a hydraulic circuit into a graph data structure does not mean formulating an independent graph structure, but does consider some portion of circuit to be a vertex and the other portion to be an arc. An algorithm visiting vertex uses the Depth First Travel method realized through recursion of function. In order to construct a node table of circuit, once check menu being selected, the HyCvtMarkNode function is called. As a function to start the Depth First Travel, it calls the HyCvtTraceLine function. Starting from a port or from a crossing point of line, it works to visit a vertex of a graph along arc. In case of meeting a port on the process of visiting, it assigns a node value to that port.

5 Vector IconEditor

Bitmap method may be used for drawing component icon, but it is inevitable to deform original shape due to inherent bitmap characteristics. To supplement these defects the vector method are adopted to store the drawing information of icon. The MainWindow of IconEditor is shown in Fig.3. For example in case of storing the line information, the start and end point informations are utilized to express drawing. The informations of the designed component consist of the component name, drawing information to express in graphic monitor, and of port information. Port information is composed of relative port position, port number, and port type. In order to design hydraulic component using IconEditor, component name must be input first and then component design progresses.

6 Conclusion

In this paper a computer software package has been developed to assist the designer of fluid power systems. Results are summarized as follows.

1. On graphic monitor icons are selected to construct system and from icon library suitable components are selected.
2. Modularized modeling techniques are presented considering connectivity between components.
3. Software techniques are developed to examine the compatibility between components together with the validity of overall system.
4. A vector IconEditor has been developed under Motif and imbedded to facilitate the addition of new components.
5. Being the X-window system based on Unix, this software has merits in portability, extensibility, and multi-user environment.

References