

공압회로 설계 및 시뮬레이션을 위한 소프트웨어 개발

손성용 신은주 이대길 곽윤근

한국과학기술대학 기계.재료공학부

Development of a design and simulation program for pneumatic systems
using computer graphics

Son Sung Yong Shin Eun Ju Lec Dai Gil Kwak Yoon Keun

School of Mechanical and Materials Engineering, KIT

Abstract

Drawing pneumatic circuits by hand and searching for the error when the circuit is not properly constructed are very difficult. In this paper, a graphic simulation program for drawing and evaluating pneumatic circuit systems was developed. The program is menu-driven style and pneumatic circuit can be easily drawn by selecting the pneumatic components from the menus. Simulation of the motion of each pneumatic component and testing of whether the circuit is constructed properly are possible with the software.

This program was written in Turbo Pascal and also used the Turbo Graphix Toolbox. The system hardware requires IBM PC, XT, AT or compatibles, and Hercules Card.

1. 서론

요즘의 여러가지 사회현상이 자동화를 더욱 필요로 하고 있으며, 또한 그 흐름에 힘입어 자동화에 대한 관심도도 높아져 가고 있다. 현재 자동화에 쓰이는 에너지원으로는 유압, 공압, 전기등이 있다. 그 중 공압을 이용한 자동화는 압축공기를 에너지원으로 하여 유용한 일을 수행하는 것이다. 그러나 어떤 목적을 수행하기 위한 공압회로의 디자인에 있어서 손으로 직접 그린다는 것은 많은 시간을 소요하는 일이며, 또한 디자인 되어진 회로가 의도하였던 대로 동작하는지에 대한 의문을 확인하기 위해서는 공압요소를 이용하여 직접 회로를 구성하여

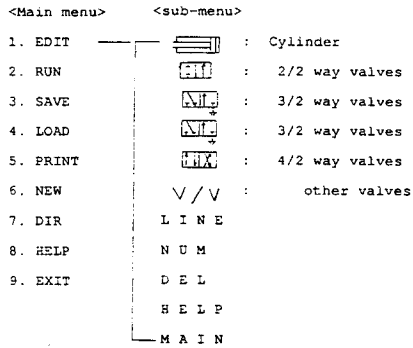
동작시켜 보아야만 한다. 그래서 본 연구는 공압회로의 디자인에 컴퓨터를 이용하여 디자인 시간을 단축하고, 실제회로를 구성해 보기 전에 시뮬레이션 시켜봄으로써 사용된 각 공압요소의 동작을 점검하여 오류 검출을 쉽게 하는데 목적을 두고 있다.

본 프로그램을 개발중 두가지의 문제점이 대두되었다. 첫째는 메모리 영역의 적절한 사용법이다. Turbo Pascal(version 3.0)은 데이터 메모리 영역이 64K로 제한되어 있다. 그래서 본 프로그램과 같이 많은 데이터를 처리해야 하는 경우 메모리 영역의 효율적인 이용이 필요하다. 둘째는 시뮬레이션시 사용된 모든 branch를 연결된 순서에 따라 순차적으로 조사(check)해야하는데 이때 많은 시간이 소요된다. 그러므로 시뮬레이션시의 시간 단축을 위해서는 branch의 효과적인 조사(check)를 위한 새로운 방법을 필요로 한다. 그래서 본 프로그램은 Grouping이란 방법을 사용하였고 이것은 만족할 만한 속도의 개선을 가져왔다.

2. 프로그램의 개요

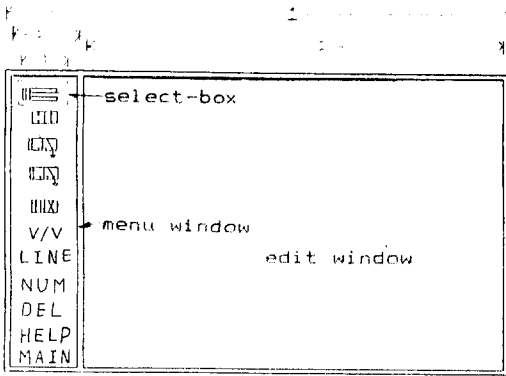
본 프로그램은 메뉴(menu-driven) 형식으로 구성되어 있다. Main menu는 9개의 명령을 가지고 있고 그 첫번째 명령인 EDIT는 11개의 sub-menu를 가지고 있다. 이 프로그램이 가지고 있는 모든 메뉴는 <Fig. 1>에 보여지고 있다.

또한 본 프로그램에서는 한 스크린면을 4개의 윈도우(window)로 나누어 사용한다. 즉 메뉴를 보여주기도 하고 또는 다른 메뉴(menu)를 선택할 수도 있게 하는 메뉴 선택공간과 공압회로 디자인을 위한 편집(editing) 공간으로 구분한다.



< Fig. 1 > Main Menu

그리고 각 윈도우마다 본 프로그램에서 사용한 Hercules Board 에 맞는 적절한 좌표값이 지정되어 있다. <Fig. 2> 는 4개의 윈도우(window) 가 구성된 모습과 각각의 좌표값을 나타내고 있다.

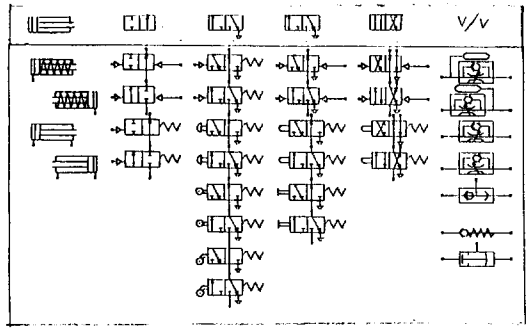


- 윈도우 1 : 전체화면 (720 × 350)
- 윈도우 2 : 편집구간 (600 × 340)
- 윈도우 3 : 메뉴구간 (95 × 340)
- 윈도우 4 : 선택 Box (80 × 30)

< Fig. 2 > 윈도우(window) 구성 모습

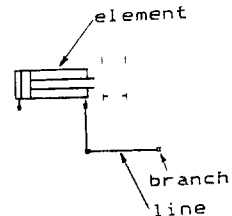
여기서, 선택 Box(select-box) 는 메뉴 선택을 위해 사용된다. 본 프로그램은 공압회로 디자인을 위해 33개의 공압요소(4개의 실린더, 4개의 2/2 way 밸브, 14개의 3/2 way 밸브, 4개의 4/2 way 밸브, 7개의 기타 밸브) 들이 심벌(symbol) 을 이용하여 메뉴 형식으로 저장되어 있으며 원하는 공압요소를 사용자가 쉽게 선택하여 사용할 수 있다. <Fig. 3> 에서는 이 프로그

램에 저장된 공압요소들을 보여주고 있다.



< Fig. 3 > 사용가능한 공압요소들

한 회로의 디자인을 위해 최대 50개의 공압요소와 255개의 branch 를 사용할 수 있으며, 모든 공압요소, branch, 연결선(line) 이 그려진 곳은 protect 를 걸어 다른 공압요소가 그 위에 겹쳐져 그려지는 것을 방지한다. 시뮬레이션(simulation) 을 위하여 공압장치(pneumatic device) 의 연결선(line) 은 각각 요소(element) 와 branch 로 나타낸다. 여기서 branch 는 각각의 요소(element) 를 연결하는 역할을 하는 교점(node) 이고, 모든 데이터는 이 branch 를 이용하여 전송된다. <Fig. 4>는 공압요소와 연결선의 예이다.



< Fig. 4 > 공압요소와 연결선(line) 의 예

배선은 수직, 수평으로만 가능하도록 되어있으므로 수직, 수평이 아닌 경우는 배선이 불가능하다. 평면에서 회로를 설계하는 것이므로 한 개의 branch 에 대해 직접 연결 가능한 branch 는 최대 4개 (상.하.좌.우)가 된다. 요소(element) 는 branch 와 연결되어 그 branch 로 부터 필요한 데이터를 받아들이고 그 데이터로 부터 동작한 결과를 구하여 다음에 연결된 branch 로 보낸다. 이러한 연결은 menu-driven 방식에 의해 컴퓨터 화면 상에서 회로를 디자인할 때 interactive 하게 이루어진다. 시뮬레이션 (simulation) 은 연속동작과 스텝동작의 두개의 option 을 가지고 있다. 연속동작은 회로에 공기가 한번 주입되면 그

에 따라 연속적으로 공압요소가 동작하는 것을 보여주고, 스텝(step) 동작은 특정한 키(key)를 누를 때마다 각 공압요소에 공기가 유입 또는 배출되며 그에 따른 공압요소들의 동작을 한 스텝(step)씩 관찰할 수 있게 한다. 각 공압요소의 연결선은 공기의 유입과 배출에 따라 굵기가 변화되므로 공기의 이동을 정확히 확인할 수 있다. 따라서 디자인된 회로의 오류를 쉽게 찾을 수 있게 한다.

3. 프로그램 개발시의 문제해결

3.1 효율적인 메모리의 이용

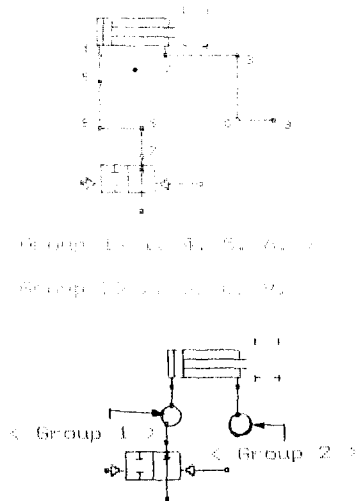
본 연구는 Turbo Pascal(version 3.0) 과 Turbo Graphix Toolbox(version 1.0)를 사용하여 IBM AT, Hercules Board 상에서 수행되었다. Turbo Pascal 은 데이터 메모리 영역이 64K byte 인데 Graphix Toolbox 가 8K byte 를 사용하므로 실제적으로 56K byte 의 메모리 영역만을 사용해야 한다는 제한이 있다. 이러한 제한조건 때문에 본 프로그램은 최대한 데이터 메모리 사용을 효율적으로 하는 방향으로 구현된다. 따라서 가능한 모든 데이터가 byte 단위로 처리된다. 특히 branch 연결 방법에 있어서 이중 연결 리스트(doubly linked list)를 사용하는 경우 시뮬레이션할 때의 속도개선 및 데이터 처리가 용이한 점이 있으나 메모리 사용상의 문제가 있다. 이중 연결 리스트를 사용하기 위해서는 포인터가 필요한데 Turbo Pascal 에서는 한개의 포인터(point)가 4 byte 씩을 차지하고 각 branch 가 4 개의 포인터(pointer)를 필요로 하므로 결국 한 branch 가 16 byte 씩을 사용 하게 된다. 그러나 1차원 배열(array)을 사용하는 경우 pointer 대신 index 가 사용되는데, 이 index 값을 byte 단위로 할 수 있으므로 한 branch 당 4 byte 가 사용된다. 즉 한 branch 당 12 byte 씩의 메모리를 절약할 수 있다. 따라서 메모리 절약을 위한 데이터 구조로 1차원 배열(array)이 사용된다.

3.2 시뮬레이션시의 시간 단축을 위한 Grouping 기법

도입

시뮬레이션(simulation) 시간을 줄이기 위하여 도입된 개념이 Grouping 이다. <Fig. 5> 회로의 예에서 실린더에 입력되어야 할 신호(signal)를 7에서 1까지 비순차적으로 연결된 branch 를 통해 전송하기 위해서는 연속되는 연결선을 7부터

1까지 (7-6-5-4-1) 추적해 나갈 필요가 있다.



< Fig. 5 > Group 의 예

이 연결선(line) 상에 branch 가 n 개 있다고 할 때 최적화된 상태로 추적해 나갈 경우 시간복잡도(time complexity)는 $O(n)$ 으로 된다. 전체의 branch 를 t 개라 하면 전체 연결선(line)의 데이터 전송을 위해서는 $O(t)$ 가 된다. 그러나 실제 회로에서 한 branch 에 연결되는 branch 가 한개만 있는 것이 아니므로 순환적(recursive) 추적이 필요하게 된다. 이는 많은 시간 낭비가 발생하고 스택(stack) 필요량이 증가하므로 효율적이지 못하다. 본 연구에서는 1차원 배열(array)이 사용되는데, 인접한 branch 와의 데이터를 비교하여 새로운 상태를 결정하는 방법을 사용하는 경우 연결선의 branch 가 순서적으로 배열되어 있지 않을 수도 있으므로 최악의 경우 n 개의 branch 를 갖는 연결선을 통한 데이터 전송을 위해서는 $O(n^2)$ 의 시간이 소요된다. 그러나 동일 연결선상의 branch 들이 같은 상태이어야 하는 점에 착안하여 한 개의 group 으로 만들어 각각의 group 을 한 개의 branch 처럼 처리함으로써 한 개의 branch 의 상태(state)를 결정하는 시간을 $O(n)$ 으로 할 수 있다. 이 경우 전체를 grouping 하는데 소요되는 시간 $O(t^2)$ 가 필요하나 grouping 은 시뮬레이션이 시작되기 전에 1회만 실행하면 되므로 일단 grouping 이 끝난 후에는 한 스텝(step)의 상태(state) 변화를 결정하기 위하여 $O(t)$ 의 시간만이 소요되므로 최적화 되었다고 볼 수 있다. 한 개의 group 은 공압요소(pneumatic element)를 한 정점(vertex)으로 할 때 두 정점간을 연결시켜 주는 작용을 하며 데이터를 전송하는 간선(edge)이라고 할 수 있다.

4. 프로그램의 구조

본 프로그램의 구조는 데이터 관리부, 그래픽(graphic) 처리부, 주처리부의 세부분으로 나눌 수 있다. 데이터 관리부는 디자인(design), loading, saving 의 기능을 가지며 시스템(system) 이 인식 가능한 형태의 내부 데이터를 만들거나, 일정한 양식(format) 으로 파일(file) 상에 관리하는 역할을 한다.

그래픽 처리부는 디자인(design) 및 시뮬레이션(simulation) 과정을 가시적으로 CRT 상에 나타내어 준다. 주처리부에서는 데이터 관리부에서 생성된 데이터를 이용하여 시뮬레이션하는 과정을 그래픽 처리하여 CRT 상에 나타낸다.

o 에디팅(Editing)

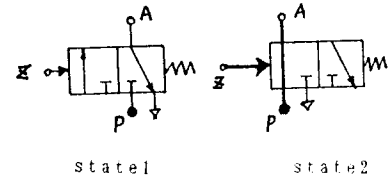
스크린(CRT) 상에서 메뉴(menu) 를 선택함으로써 간단히 회로(circuit) 를 구성할 수 있다. 그래픽과 메뉴를 사용한 대화식 설계(design) 방식이므로 복잡한 Format 를 사용할 필요가 없고 단지 몇개의 키(key) 를 사용하여 선택(selection) 함으로써 공압회로의 설계를 가능하게 한다.

o 그래픽(Graphic)

Turbo Graphix Toolbox(version 1.0) 을 사용하여 공압요소를 심벌(symbol) 로 나타내어 그래픽을 처리한다. 각 공압요소가 공기의 유입과 배출에 따라 최대 2개의 상태(state) 를 나타낼 수 있게 함으로써, 실제 회로의 동작에 가까운 시뮬레이션을 할 수 있도록 modeling 한다. 모든 공압요소는 에디팅시에는 안정(stable) 한 상태로 나타난다.

o 주처리부

본 연구에서는 메모리 사용문제와 실행시간, 그래픽(graphic) 처리 문제를 고려하여 해석적인 방법으로 공압요소의 동작을 시뮬레이션 하지 않고 on/off 두개의 상태(state) 로 시뮬레이션한다. <Fig. 6> 에서 보는 바와 같이 공압요소(element) 는 입력되는 branch(z, p) 의 데이터를 받아들여 새로운 상태로 변환한 후 출력 branch (A) 의 보조기억장치(co-state) 로 데이터(data) 를 전송한다. 이 데이터로부터 다시 branch 의 상태(state) 를 결정하는 과정을 반복함으로써 폐환(feed back) 회로의 해석을 가능하게 한다.



step	Z	co-Z	power	A	co-A	element state
1	0	0	1	0	0	1
2	0	1	1	0	0	1
3	1	1	1	0	0	1
4	1	1	1	0	0	2
5	1	1	1	0	1	2
6	1	1	1	1	1	2

1. 초기치(initial state)
2. branch Z 에 데이터 전송
3. branch Z 의 상태(state) 결정
4. 공압요소(valve) 의 상태(state) 결정
5. branch A 의 보조기억장치로 output
6. branch A 의 상태(state) 결정

< Fig. 6 > 3/2 way 밸브에 공급구(power line) P 에 power 가 들어와 있는 상태에서 제어구(control line) Z에 의해 power가 작업구(working line) A 로 출구되는 예

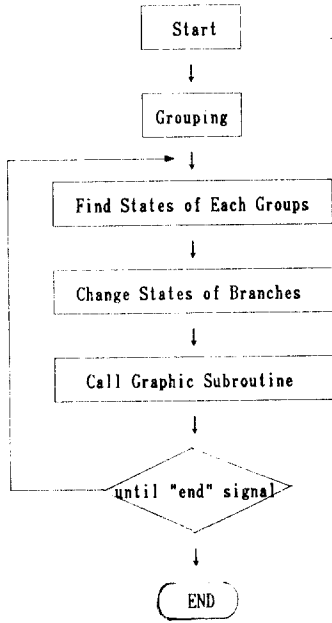
무방향 그래프(undirectional graph) 에 의한 데이터 구현으로 동일한 동작을 하여야 하는 branch 들을 한개의 group 으로 묶어서 실행시간을 줄인다. 또한 그래픽 서브루틴(sub-routine) 과 결합시켜 회로의 변환과정을 실시간(real time) 으로 나타냄으로써 설계한 공압회로의 검사(check) 를 가능하게 한다.

5. 결 론

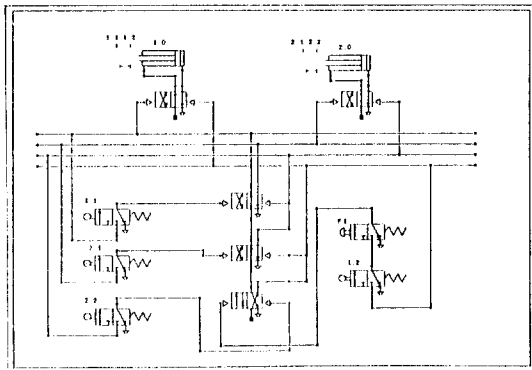
본 연구에서 공압시스템(pneumatic system) 을 위한 CAD software 를 개발하였다. 이를 사용하여 공압요소가 12개인 <Fig. 8> 의 회로를 설계하여 시뮬레이션 시켰을 때 1 cycle,

28 step 의 동작에 45 sec 가 소요되었다. 이는 실시간(real time) 동작에 거의 가깝다. 또한 < Fig. 8 > 의 예제외에 다른 회로들을 설계하여 봄으로써 케환 (feed back) 처리에 대해서 충분한 신뢰성을 증명하였다.

본 프로그램은 공압을 이용한 자동화 교육을 위해 공압장비를 구입할 수 없는 경우에도 실제회로를 구성하여 실험한 것과 같은 교육효과를 얻을 수 있으며, 간단한 산업용 설계에도 유용하게 사용할 수 있다.



< Fig. 7 > 주처리부의 Flow Chart



< Fig. 8 > 예제회로

Reference

- (1) "Turbo Pascal", BORLAND INTERNATIONAL Inc., 1983
- (2) "Turbo Graphix Toolbox", BORLAND INTERNATIONAL Inc., 1985
- (3) 이석호역, "자료구조론", 홍릉과학출판사, 1987
- (4) Robert Jourdain, "Programmer's Problem Solver for the IBM PC, XT & AT", Brady Communications Company Inc., 1986
- (5) "공압기습입문", 한국 FESTO 주식회사, 1978