

PC를 이용한 수평다관절형 로봇의 대화식 작업장 설계  
(The Interactive Robot Work-Cell Layout for SCARA type using PC)

최기봉\*      국금환\*  
(K.B. Choi    K.H. Kuk)

\* 한국기계연구소 로봇공학실 (KIMM Robotics Lab.)

This study is to develop the interactive program for work-cell layout of SCARA type robot using PC.

To make the program interactively, we made use of the software which have the function of menu, dialog box, and graphic. By using the mouse, we can progress the program quickly and easily. It has 2½ dimension and provides that one can layout the elements(robot, peripheral device, and etc.), create the work path, and calculate the robot cycle time. By comparing with cycle times of any number of work-cell layout for the same work, we can evaluate work-cell layout and select one work-cell layout to be considered optimal one which has the least cycle time.

1. 서론

용접 또는 조립작업 등에 로봇트를 사용하는 경우 생산성 향상을 위하여 로봇트 작업장의 layout를 고려해야 한다. 로봇트 작업장의 layout을 위하여 GALOP/2D, PLACE 등의 프로그램이 개발되었다 /1/2/3/4/.

본 연구에서는 조립작업을 대상으로 수평다관절형 로봇트의 작업장을 작업자가 PC의 화면을 보면서 주로 mouse를 이동시켜 가며 대화식으로 작업을 진행시킬 수 있는 프로그램을 개발하는 것이다. 즉 menu, dialog box, graphic 기능이 있는 소프트웨어를 이용하여 모든 작업점들이 로봇트의 작업영역 안에 포함될 수 있도록 로봇트와 주변기기등의 PC상의 도형을 신속하게 이동 및 회전시키며, 완성된 동일 작업의 다수의 작업장을 상호 비교 및 검토할 수 있게 한다. 이를 위하여 PC의 화면상에 요소(로봇트, 주변기기 등)들을 생성시키고, 적절한 위치에 배치한다. 그후 로봇트가 작업할 작업점들 사이의 경로를 결정 한 후, 주어진 경로를 따라 로봇트가 작업했을 때 소요되는 cycle time을 계산한다. 작업자는 위의 방법으로 계산된 cycle time으로 동일 작업의 여러 형태의 작업장을 서로 비교 및 검토하여 최적이라 고려되는 작업장을 선택한다. 최종적으로 작업자는 작업장 내에 있는 요소들의 절대 위치와 회전각도 등의 데이터를 얻을 수 있다.

위의 작업을 하기 위해 PC의 화면상에 생성되는 도

형들은 요소들의 수평절단면이며, 경우에 따라 작업점의 수직위치도 알 수 있어 프로그램은 2½차원을 갖는다.

2. 프로그램이 갖추어야 할 조건

본 연구에서는 기구학적 구조로써 현재 많이 이용되고, 또한 계속적인 이용 증대가 예상되는 수평다관절형 로봇트의 작업장 layout을 위한 프로그램 개발을 구체적인 적용범위로 결정하였다. 그리고 개발한 프로그램의 이용, 성능과 설계의 관점에서 아래와 같은 프로그램의 구비조건을 결정하였다.

- ① PC(Personal Computer) 상에서 이용될 수 있어야 한다.
- ② Layout 작성과 평가가 PC의 화면상에서 신속하고, 용이하게 수행될 수 있어야 한다.
- ③ 프로그램의 이용자가 프로그램 자체에 대한 지식이 없어도 단지 대화식 작업에 의해서 프로그램의 이용이 가능해야 한다.
- ④ 로봇트 작업장을 구성하는 computer model은 layout 작업에 요구되는 모든 기하학적, 물리적, 기술적 정보를 가져야 한다.
- ⑤ 개발한 프로그램의 유연성을 높이기 위해서 작업장 요소들의 computer model 생성과 이용처리를 상호 분리시킨다.
- ⑥ 프로그램의 개발과 미래의 확장을 보다 용이하게

실행시키기 위해 프로그램을 module 구조로 설계한다.  
 ⑦ computer를 이용하지 않는 기존의 layout 작업이 PC를 이용하여 수행될 수 있도록 화면상의 보조 작업 요소가 그려되어야 한다.

### 3. 프로그램의 구조

본 프로그램은 Microsoft Windows의 환경에서만 실행될 수 있으며, Windows 개발 tool을 사용하여 C 언어로 작성하였다 /5//6/. 그러므로 Windows 개발 tool에서 제공하는 menu, dialog box 및 graphic 기능들이 이용되었으며, 특히 mouse를 사용하여 이 기능들을 용이하게 다룰 수 있게 하였다.

본 프로그램은 menu를 선택함으로써 작업을 진행시킬 수 있으며, 사용된 주 menu와 부 menu의 구조는 (그림 1)과 같다. 주 menu는 menu bar에 의해 화면상에 항상 고정되어 있으며, 부 menu는 주 menu를 선택할 때 발생하는 pull-down menu로 되어있다. pull-down menu의 각 항목을 선택하면 그 항목에 해당하는 내용의 작업을 진행시키기 위한 dialog box가 (그림 2)와 같이 화면상에 출력되며, 작업자는 dialog box의 각 입력란에 mouse 또는 key에 의해 입력함으로써 대화식으로 작업을 진행시킨다.

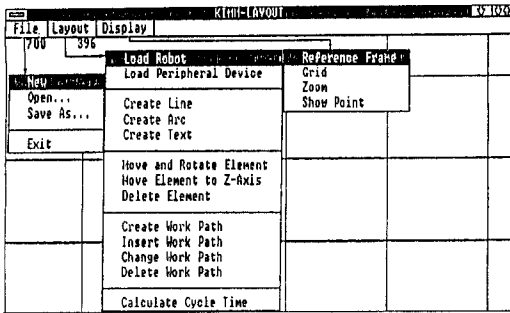


Fig. 1 Structure of the main and the sub menus

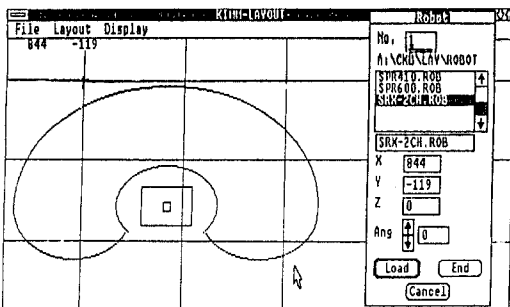


Fig. 2 Input processing by using dialog box

각 menu에서 작업할 수 있는 내용을 간단하게 서술하면 다음과 같다.

#### (1) File

로봇 작업장의 data file을 처음 생성하거나, 완성된 작업장의 data를 file에 저장시킬 때 사용한다. 또 프로그램의 종료시킬 때에도 사용한다.

#### (2) Layout

##### ① Load Robot / Load Peripheral Device

미리 file로 만들어 놓은 로봇이나 주변기기들의 데이터를 읽어들이어 화면상에 데이터에 해당하는 도형을 출력시킨다. 이들 요소의 데이터는 double linked list로 서로 연결되어 있어 데이터를 동적으로 추가 및 삭제가 용이하다.

로봇의 데이터 구조는 기하학적 데이터 이외에 이동경로에 소요되는 cycle time을 계산하기 위한 필요한 데이터(로봇 각축의 최대속도)로 이루어져 있으며, 화면상에 출력되는 도형은 로봇의 수평 작업영역이다.

주변기기는 premitive로 서로 연결된 구조를 채택하였으며, 주변기기에 작업점의 위치를 표시할 수 있게 하였다.

##### ② Create Line / Create Arc / Create Text

작업장 설계를 위한 보조도구인 직선, 원호 및 text를 생성한다.

이들 요소의 데이터도 로봇, 주변기기와 함께 double linked list로 연결되어 있다.

##### ③ Move Rotate Element / Move Element to Z-Axis / Delete Element

요소들의 이동, 회전 및 삭제를 할 때 사용한다. 각각의 요소들은 기준점을 가지고 있으며, 화면상의 기준점 위치에 조그만 사각형이 나타나 있어 그 점을 mouse로 선택하여 dialog box에 입력함으로써 작업을 진행시킨다.

##### ④ Create Work Path / Insert Work Path / Change Work Path / Delete Work Path

로봇이 작업할 이동경로를 생성, 추가, 변경 및 삭제할 때 사용한다.

작업경로를 생성할 때에 작업할 로봇과 작업점들을 mouse를 이용하여 dialog box에 순서대로 입력하면 입력 순서에 따라 작업점들 사이가 직선으로 연결되어 화면에 출력되기 때문에 작업자가 작업경로를 확인하면서 작업할 수 있다.

##### ⑤ Calculate Cycle Time

작업경로를 생성한 후에 이동경로를 위해 선택된 작업점들이 로봇의 작업영역 안에 있는지를 확인하고, 작업경로를 따라 이동했을 때 걸리는 cycle time을 계산한다.

#### (3) Display

① Reference Frame

기준좌표의 설정 및 작업장의 크기를 결정한다.

② Grid

grid의 생성, 제거 및 grid의 크기를 결정한다.

③ Zoom

작업장 크기를 확대 또는 축소하여 화면상에 출력한다.

④ Show Point

각각의 요소들은 mouse로 선택하기 위한 기준점을 가지는 데 이 기준점은 기준점의 위치에 조그마한 사각형으로 화면상에 출력되어 나타나 있다. 이 기준점이 필요 없을 때에는 제거하거나, 제거된 후 또 다시 출력할 때 사용한다.

4. Cycle Time 계산

로봇이 조립작업을 할 때에는 기본적으로 gross motion, interface motion 및 fine motion의 3단계 motion을 한다 /7/.

gross motion은 로봇이 고속·저정밀도를 가지고 목표점을 향해 이동하고 있는 motion이고, interface motion은 gross motion에 비해 상대적으로 저속·고정밀도로써 목표점에 근접하는 motion이며, fine motion은 실제 목표점과 접촉할 동안의 motion이다.

조립작업시 로봇의 cycle time을 계산할 때 gross motion 동안의 시간(GMT)은 로봇의 사양에 의해 알 수 있으나 interface motion과 fine motion 동안의 시간(FIMT)은 추정해야만 한다 /7/.

한 작업점에서 다음 작업점까지 로봇이 이동할 때 GMT의 계산은 다음과 같다.

$$GMT = \max(T_1, T_2) + T_z$$

여기서  $T_i$  :  $i$ 축이 최대속도로 이동했을 때 걸린 시간

그러므로 로봇의 cycle time  $T$ 는

$$T = \sum (GMT_i + FIMT)$$

이다.

로봇이 각 점들 사이를 이동할 때에는 항상 가·감속을 해야만 한다. 그러므로 각 축이 최대 속도로 이동할 때 걸린 시간을 계산할 때에는 가·감속 시간을 고려해야 한다.

가·감속을 할 때 시간에 따른 속도형태는 (그림 3)과 같이 2가지로 나눌 수 있다.

(그림 3-a)와 같이 가속, 등속 및 감속 구간을 거칠 때 걸린시간  $t_2$ 는

$$t_2 = t_1 + T_m$$

여기서  $t_1$  : 가·감속을 고려하지 않았을 때 걸린 시간

$t_2$  : 가·감속을 고려했을 때 걸린 시간

$T_m$  : 가속 또는 감속시간

이고, (그림 3-b)와 같이 이동 시간이 너무 짧아 등속 구간을 거치지 않을 때 걸린시간  $t_2$ 는

$$t_2 = 2 \sqrt{t_1 \cdot T_m}$$

이다.

그러므로  $T_i$ 를 계산할 때에는 위의 식을 고려한다.

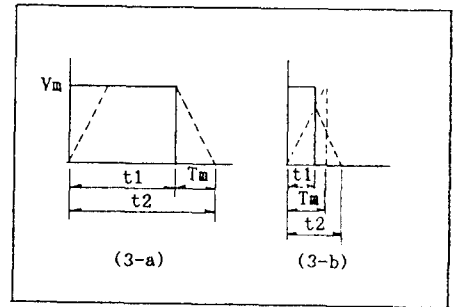


Fig. 3 Velocity profile with acceleration and deceleration

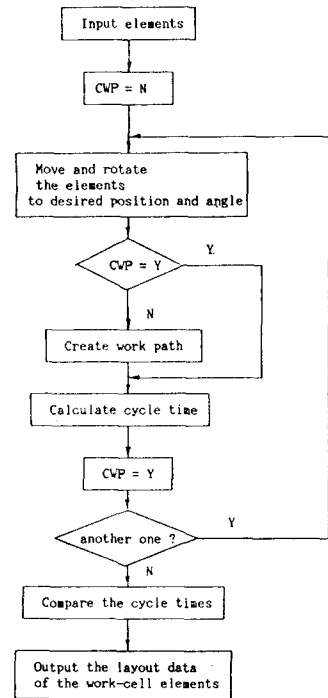


Fig. 4 Robot work-cell layout processing flow

5. 프로그램의 실행 예

작업장 layout을 위한 데이터를 얻기 위해서는 (그림 4)와 같은 과정을 거친다. 먼저 로봇, 주변기기 등의 요소를 dialog box를 통해서 읽어 들인 후 원하는 위치와 각도로 mouse를 사용하여 배치한다. 그후

작업경로를 결정한 후 cycle time을 결정한다. 그 다음은 작업자의 판단에 의하여 요소들을 재 배치하여 동일 작업의 또다른 작업장을 만들거나, 그다음 단계로써 동일 작업의 여러 작업장의 cycle time을 서로 비교한다. 요소들을 재 배치할 때에는 작업경로가 이미 결정되어 있으므로 또 다시 경로를 결정할 필요는 없다. 최종적으로 작업자는 여러 cycle time을 서로 비교하여 제일 작은 cycle time을 갖는 작업장을 선택하여 그것의 데이터를 얻으면 된다.

다음은 실행 예로써 다음과 같은 조립작업을 고려한다. 입력 buffer에 palette이 입력되면 palette을 feeder들 사이로 이동시켜 feeder에서 출력되는 작업물들을 palette에 미리 지정된 위치에 옮겨 놓은 후 palette을 출력 buffer로 출력시키는 작업이다. 이와 같은 작업의 각 요소들의 배치와 이동경로의 생성은 (그림 5)와 같다. 여기에서 각 작업점들 사이가 직선으로 나타나 있는데 이것은 실제 로봇의 이동경로와는 무관하며 다만 한 작업점에서 다른 작업점에서의 이동만을 표시했을 뿐이다. 이 작업에서 cycle time 계산을 위해 FIMT 는 0.2 sec  $T_m$  은 0.3 sec로 놓았다. cycle time 계산에서 사용된 각 축의 각도와 이동 시간 및 작업의 cycle time은 (그림 6)과 같으며, 이 cycle time을 갖는 작업장의 각 요소 배치 데이터는 (그림 7)과 같다.

## 6. 결과 및 고찰

수평다관절형 로봇트 작업장을 PC의 화면상에서 신속하고, 용이한 방법으로 layout 및 비교 검토할 수 있는 프로그램을 개발하였다. 구체적인 내용은 다음과 같다.

- ① layout 작업을 menu와 dialog box를 이용하여 대화식으로 작업이 가능하며, mouse만을 사용하여 거의 모든 작업을 수행할 수 있으므로 신속하고, 용이하게 작업을 진행시킬 수 있다.
- ② 동일 작업을 위한 다수의 작업장의 cycle time을 서로 비교하여 최적이라 고려되는 작업장을 선택할 수 있다.

본 프로그램은 현재 계속하여 개발 중에 있으며, 앞으로 로봇트의 실제 거동을 확인할 수 있는 simulator를 더 추가할 예정이다.

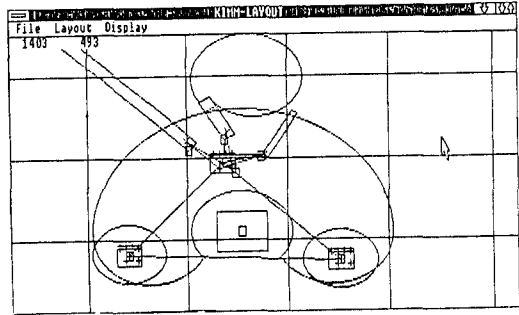


Fig. 5 Example of the Robot work-cell layout and created the work path

the1	the2	Time1	Time2	TimeZ	Time
133.372	-80.314				
-25.466	113.090	1.114	1.474	0.970	2.644
33.673	-71.685	0.603	1.422	0.970	5.235
-32.267	120.871	0.638	1.469	0.970	7.874
52.327	-69.421	0.733	1.455	0.970	10.499
-23.279	118.480	0.687	1.440	0.970	13.110
22.070	-102.711	0.528	1.643	0.970	15.922
-30.673	134.897	0.569	1.560	0.970	18.652
22.070	-102.711	0.569	1.560	0.970	21.382
-26.766	134.041	0.548	1.555	0.970	24.107
22.070	-102.711	0.548	1.555	0.970	26.832
-22.832	132.758	0.525	1.547	0.970	29.549
22.070	-132.711	0.525	1.547	0.970	32.266
-18.851	101.055	0.502	1.537	0.970	34.973
22.070	-132.711	0.502	1.537	0.970	37.680
-25.466	113.090	0.541	1.610	0.970	40.460
-143.872	35.608	0.907	0.357	0.970	42.537
133.372	-80.314	1.721	1.368	0.970	45.428

Cycle Time = 45.428

Fig. 6 The calculated cycle time

```

PALETTE
X = 594.0, Y = 329.0, Z = 350.0, Ang = 90.0

FEEDER-3
X = 634.0, Y = 411.0, Z = 350.0, Ang = -30.0

FEEDER-2
X = 406.0, Y = 441.0, Z = 350.0, Ang = 45.0

FEEDER-1
X = 550.0, Y = 493.0, Z = 350.0, Ang = 30.0

SPR600
X = 619.0, Y = 38.0, Z = 0.0, Ang = 0.0

BUFFER
X = 1009.0, Y = -104.0, Z = 350.0, Ang = 90.0

BUFFER
X = 166.0, Y = -74.0, Z = 350.0, Ang = 90.0

```

Fig.7 The element data of an robot work-cell layout

## 참고문헌

1. R.Jayarman, "GALOP/2D:A graphical system for work-cell layout evaluation".
2. Stuart J.Kretch, "Robotic animation", Mechanical engineering, August, 1982.
3. 최 진섭, "로봇트 작업장 배치를 위한 로봇트 작업 영역에 관한 연구", M.S.Thesis, KAIST, 1988.
4. 정 재운, "그래픽 시뮬레이션을 이용한 로봇트 용접 공정 계획", M.S.Thesis, KAIST, 1984.
5. Microsoft Windows User's Guode Ver.2.0
6. Microsoft Windows Software Development Kit Programmer's Learning Guide Ver.2.0
7. Anthony S. Kondoleon, "Cycle time analysis of robot assembly systems".