技術論文 한국정밀공학회지 제14권 제5호 (1997년 5월)
Journal of the Korean Society of Precision Engineering Vol. 14. No. 5. May. 1997.

선체 블록 용접을 위한 효과적 로봇 오프-라인 자동교시 소프트웨어 개발 연구

임 생 기*, 최 재 성**, 홍 석 관**, 한 용 섭***, 범 진 환****

Automatic Offline Teaching of Robots for Ship Block Welding Applications

Seang Gi Lim*, Jae Sung Choi**, Sok Kwan Hong**, Yong Seop Han***, Jin Hwan Borm****

ABSTRACT

Computer aided process planning and Offline programming are decisive factors in successful implementation of automated robotic production. However, conventional offline programming procedure has proven ineffective due to time-consuming teaching process for robot programming and due to inefficient system modeling. The paper presents an efficient procedure to semi-automatically generate robot job programs for ship block welding applications. In the research, the teaching positions are automatically determined by predefined rules which are functions of the type and the dimensions of the given welding section of ship block. And a sequence of robot movements and welding conditions such as welding type, welding current, welding speed, and welding torch orientation, are determined by use of Standard Program which is experimentally proved to work well for the welding section group. Finally, a robot program for the welding section is generated automatically. Based on the algorithm, a offline automatic teaching software is developed. The paper presents also the algorithm and structure of the software.

Key Words: Offline programming(오프라인 프로그래밍), Position determination Rule(위치결정룰), Standard Program(표준프로그램), Automatic Programming (자동 프로그래밍)

1. 서 론

최근 조선 산업에서 로봇을 이용하여 선체를 이루는 블록(block)들 내부의 용접을 자동화하려는 노력을 하고 있다. 배 한 척을 건조하기 위해서는 Fig.1 에서와 같은 형상의 블록들이 약 380 여개(VLCC double deck 인 경우)가 소요되며, 각각의 블록에는 횡(trans.)부재와 종(longi.)부재가 서로 만나서 만들어지는 Fig.2 와 같은 용접대상 부재(section)가 약 200 여개가 존재하는데, 이들을 로봇을 이용하여 용접하려는 시도가 최근 활발히

* 아주대 시스템공학과 대학원
** 고등기술연구원 생산기술연구실
*** 대우중공업 선박해양연구소
**** 아주대학교 기계및 산업공학부

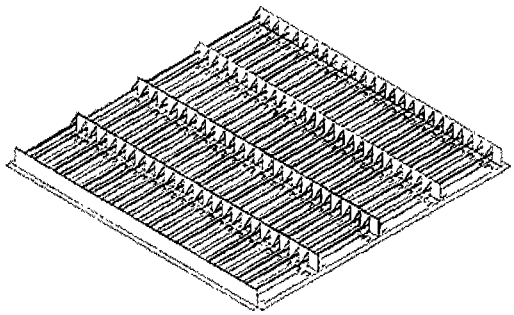

Fig. 1 Typical Ship Block

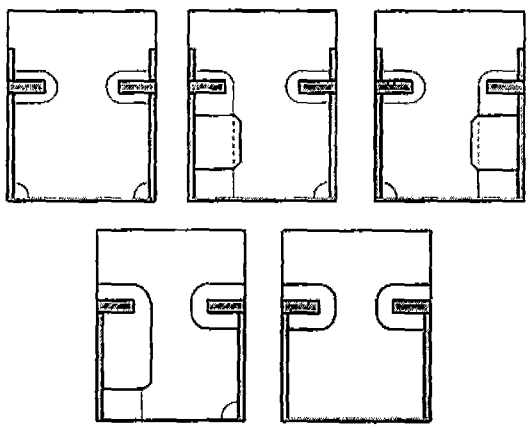

Fig. 2 Types of Welding Sections

추진중이며, 그 방법들이 연구되고 있다. 하지만, 조선 산업에서는 그 특성상 규격화된 제품이 극히 제한되어 있고 대부분 그 크기나 종류가 서로 다른 부품들로 구성되어 있기 때문에 로봇을 이용하기 매우 어렵다. 즉, 상기의 블록에서의 200 여개의 용접 대상물 등은 용접선 형태나 종류는 제한적이지만, 그 크기나 위치는 각각 다를 수 있으며, 또한 용접선의 형태와 종류에 따라 각종 용접 조건(용접 속도, 기울기, 전압 및 전류 등)또한 달라져야 한다는 것이다. 이는 각각의 작업 대상물마다 로봇 프로그램이 각각 달라져야 함을 의미하며, 이 경우 로봇 프로그래밍 하는 데 걸리는 시간이 실제 용접하는 시간보다도 훨씬 더 많이 소요됨을 의미한다.

오프-라인 프로그래밍(off-line programming) 방법은 상기문제를 해결해 줄 수 있는 실마리를 제공하고 있으며, 이와 관련된 연구가 이미 수행되어진 바 있다[1][2]. 대표적인 오프-라인 프로그래밍용 소프트웨어로 Deneb 사의 IGRIP(interactive graphics robot instruction program)과 Tecnomatix 사의 ROBCAD 등이 있으나, 이들은 범용이기 때문에 이를 사용하여 로봇 프로그램을 생성해야 할 경우, 용접 대상물이 바뀔 때마다 CAD 도구를 이용하여 그 대상물의 형상 모델링 작업을 다시 수행하여야 하고, 로봇이 지나가야 할 모든 작업 점들을 다시 정의 내려야하는 등 많은 시간과 노력이 든다. 그 뿐만 아니라, 오프-라인 프로그래밍 응용시 필수적으로 수행해야 하는 로봇 캘리브레이션의 결과 데이터나 용접 조건 등 경험적/실험적 데이터를 쉽게 반영하기 어렵게 되어 있다.

이에 본 연구에서는 범용 오프-라인 소프트웨어를 기반으로 하여 아아크 용접 로봇 시스템이 선체 대조립 용접에 효율적으로 사용될 수 있도록, 로봇 오프-라인 자동 교시(robot off-line automatic teaching) 시스템을 개발하였으며, 본 논문에서는 이의 구성, 알고리즘 및 실행결과 등을 소개하고자 한다. 본 오프-라인 자동 교시 시스템은 아아크 용접기를 장착한 로봇들을 이용하여 선체 대조립 공정이라는 특정 응용에 국한되지만, 프로그램 응용시 자주 수행하여야 하는 로봇시스템 캘리브레이션의 자동화를 이룩하였으며[5], 용접 대상물의 형상 데이터, 캘리브레이션 데이터, 용접선 형태에 따르는 용접 조건이나 로봇 동작 등을 포함하는 표준 프로그램 데이터 등 경험적/실험적 데이터 베이스 구축이 매우 손쉽고, 그 데이터 베이스를 바탕으로 수십 대의 로봇 시스템이 수백 개의 치수가 서로 다른 용접 대상물을 용접할 수 있는 로봇 프로그램들(200 개정도)을 짧은 시간 내에 자동 생성할 수 있으며, 생성된 프로그램들 각각을 이용하여 각종 시뮬레이션을 수행하여 로봇의 동작, 다른 물체와의 충돌 등을 확인할 수 있다. 본 시스템은 현재 조선 현장에 성공적으로 실험 가동중이다.

2. 로봇시스템 운영 시나리오

본 오프라인 자동 교시 소프트웨어가 이용될 목적 하드웨어(Target Hardware) 는 Fig. 4 ,Fig.5 및 Fig. 10 등에서 보여주는 것과 같이 아아크 센서 및 접촉센서가 장착된 미니급 6 축 다관절 로봇과 이 로봇이 설치되는 카트(Cart) 로 구성되어 있다. 또한, 작업 현장에는 수십

대의 로봇시스템을 줄로 매달아서 상하좌우로 이동시킬 수 있는 크레인이 설치되어 있어서, 이를 이용하여 수동으로 각각의 Cart 들을 용접대상 부재에 설치하도록 하였다. 로봇이 설치된 Cart 에는 3 개의 막대기(Bar) 가 있으며 그 막대기 끝이 용접대상물 부재의 각 면에 접촉하도록 Cart 를 용접대상물에 설치하도록 한다. 이 작업은 수동으로 수행되어지지만 그 설치오차는 로봇 손끝 오차가 보통 작업시 5 mm 이상 되지 않음이 확인되었다. 이와 같이 수동으로 로봇을 용접대상물에 설치하는 작업은 시간도 많이 걸리고 오차도 많기 때문에 위치 센서를 이용하여 온 라인으로 보정하는 연구가 진행 중에 있다. 상기와 같이 설치된 로봇 시스템과 용접대상물 상의 용접선과의 모델링위치와 실지 위치와의 오차는 여러 가지 이유(로봇 설치오차, 용접대상물 제작 오차 등등)로 말미암아 매우 클 수 있다. 본 시스템의 경우, 로봇/Cart 시스템 캘리브레이션을 수행하였다 하더라도 최대 8 mm 정도의 전체적 오차를 발견할 수 있었다.

따라서, 정확한 용접 시작점과 끝점의 위치는 접촉 센서를 이용하여 알아내며, 용접선을 정밀히 추종하기 위해 아아크 센서를 이용한 추종 알고리즘을 이용하고 있다. 단, 본 오프라인 자동 교시 소프트웨어는 접촉 센싱을 위한 초기 위치들을 용접대상물의 종류 및 크기에 따라 자동으로 계산해 주며, 용접선 형태에 따르는 용접 조건 등을 설정해 주는 것이다. 이 과정을 간단히 요약하면 아래와 같다. 먼저, 각 용접대상물 부재의 종류별로 용접선 형태에 따르는 용접 조건/속도 및 로봇 동작 과정 등을 포함하는 표준 프로그램을 특정 크기를 갖는 표준용접 부재에 대하여 실험을 통해 만들어 놓는다. 같은 종류이지만 크기가 표준 용접 부재와 다를 경우 본 오프라인 자동 교시 소프트웨어가 그 교시점을 계산하여 준다.

예를 들면, Table 3 는 본 오프라인 자동 교시 소프트웨어의 최종 결과이지만, Table 3 의 아래 부분의 교시점 데이터만을 제외하고는 표준 프로그램과 동일하다. 즉, 본 오프라인 자동 교시 소프트웨어가 작동하기 위해서는 각 용접대상 부재(Section) 형태별로 표준 부재에 대한 증명된 로봇 프로그램을 사전에 실험을 통해 만들어 놓아야 하며, 이것이 표준 프로그램인 것이다.

3. 시스템 구성

본 오프-라인 교시 시스템의 기능상 구성과 데이터의 흐름은 Fig.3 와 같으며, 크게 세 부분으로 나누어져 있다. 즉, 제안된 시스템은 먼저, 로봇 시스템의 정확한 모델 파라미터를 얻기 위한 캘리브레이션 부분과 데이터 베이스(data base) 구축을 위한 부분(config.)과 이 데이터 베이스를 이용하여 로봇 프로그램을 자동 생성해주고

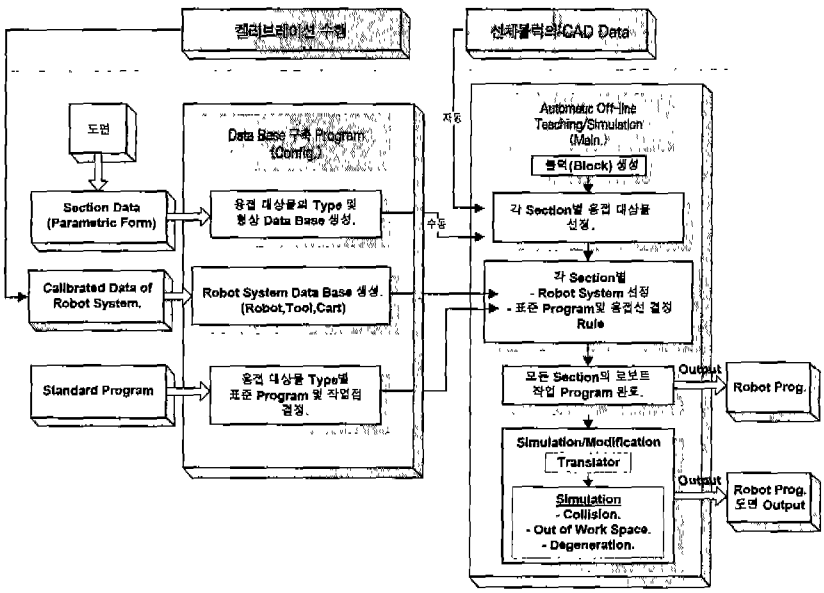

Fig. 3 Schematic Flow of the Offline Programming System

각종 시뮬레이션을 수행할 수 있는 부분(main.) 으로 구성되어 있다.

〔config.〕 부분은 외부로부터 들어오는 오프-라인 프로그래밍 시스템에 필요한 각종 데이터들을 받아들여 이를 정리하는 과정이 된다. 여기서, 외부에서 입력되는 데이터에는 작업 대상물 치수(section data), 로봇 시스템의 캘리브레이션 데이터(calibration data of robot system), 그리고 로봇의 표준 프로그램 (standard program)과 그 프로그램의 교시 데이터(teaching data)이다. 작업 대상물 치수(section data)는 용접 대상물의 형상 데이터로 이는 작업 대상물의 형태를 결정하며, 표준 프로그램의 교시 데이터와 함께 새로운 로봇 프로그램을 생성하는데 있어서 가장 중요한 데이터 중 하나이다. 캘리브레이션 실험을 통해 얻어진 모델 파라미터(로봇의 위치, 로봇의 각 링크들의 파라메터들, 손끝의 위치)값들은 오프-라인 프로그래밍을 위한 컴퓨터상의 로봇 시스템, 작업 대상물의 시스템 모델 및 기구학적 관계를 구축하는데 이용되어 진다. 로봇 표준 프로그램은 각 용접대상 부재(Section) 형태별로 표준 부재에 대한 증명된 로봇 프로그램을 말하며, 여기에는 용접대상 형태별 각종 경험적 데이터를 포함하는 작업 조건/속도 및 작업 시퀜스가 포함되어 있는 셈이다.

Table 3 는 본 오프라인 자동 교시 소프트웨어의 최종 결과이지만, 표준 프로그램 역시 교시점 데이터만을 제외하고는 이와 동일하다. 이 로봇 표준 프로그램은 작업대상물 종류(Types) 에 따라 서로 다른 형태로 여러 개를 생성할 수 있으며, 이것은 한 종류의 작업 대상물 모양과 치수가 변화하더라도 경험적 표준 작업 조건 및 작업 시퀜스를 유지하도록 할 수 있게 한다.

〔main.〕부분은 〔config.〕 상에 이미 구축된 데이터 베이스를 선택하여, 실제 작업을 하는 로봇에 필요한 작업점을 계산 하므로 서 로봇 작업 프로그램을 자동 생성하도록 하는 과정이다. 이 과정에서는 이미 구축된 작업 대상 블록, 이 블록을 용접 할 캘리브레이션된 로봇 시스템, 작업조건 및 작업 과정 등이 명시된 표준 프로그램을 선택하게 된다. 즉, 작업 대상물의 형상 데이터, 로봇 시스템을 정의하는 로봇의 작업대상물과의 상대적 위치, 로봇의 각 축의 링크 파라미터들, 손끝의 위치 데이터 등의 캘리브레이션 데이터, 용접대상물에 알맞는 용접조건 및 작업 과정(Sequence) 등을 선택하는 것이다. 이와 같이 용접대상물, 로봇 시스템 및 표준프로그램 선택이 완료되면, 프로그램은 작업 대상물의 크기변화에 따르는 작업 시퀜스상의 새로운 작업점을 계산하고, 이 작업점으로 로봇을 정밀히 제어하기 위한 캘리브레이션된 역기구학(calibrated inverse kinematics) 해를 구하여, 선택된 표준 프로그램상의 작업조건과 작업 시퀜스와 동일하지만 각각의 작업점 위치는 선택된 용접대상물에 알맞게 재계산되어지게 된다. 또한 새로운 프로그램이 원만히 작동되는 지를 시뮬레이션해 볼 수도 있다. Fig. 4는 본 S/W가 여러 대의 로봇 시스템을 위한 로봇 프로그램들을 동시에 생성하여 각각 어떻게 분배되는 지를 보여주고 있다.

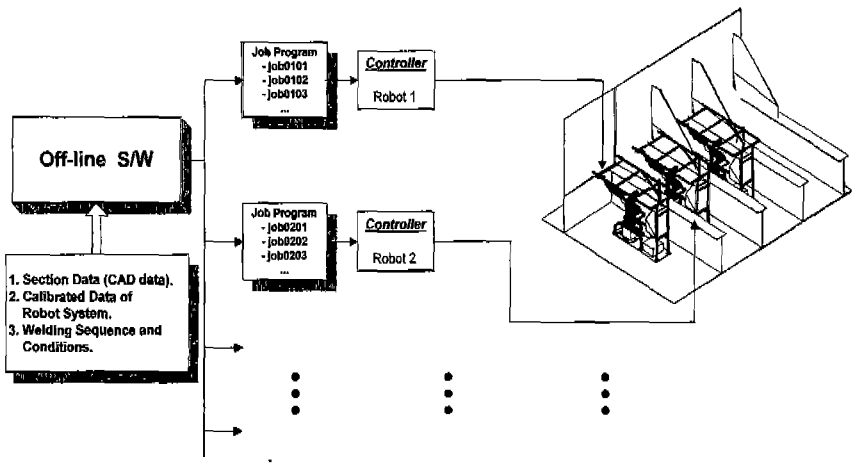

Fig. 4 Flow of distributing Robot Programs generated through the Offline Programming System

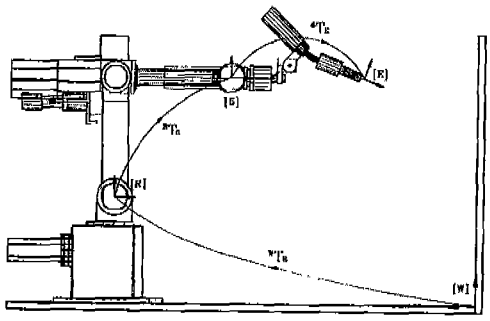

Fig. 5 Coordinates for defining Robot System

4. 로봇 시스템 모델링.

본 S/W 에서 사용되고 있는 각종 좌표계와 그들 사이의 변환 행렬 및 그들을 정의하는 각종 파라미터를 간단히 언급하고자 한다. 본 로봇 시스템을 정의하기 위하여, Fig. 5 에서 보는 바와 같이 용접 대상물에 위치한 기준 좌표계[W], 로봇의 베이스 좌표계[R], 로봇의 6축 좌표계[6], 그리고 로봇 손끝의 좌표계[E] 를 정의하였고, 그들 사이의 좌표변환 및 정의된 파라미터는 아래와 같다.

(1) 베이스 변환(base transformation) $({}^{W}T_{R})$

실제 작업상에서 용접 대상물의 기준 좌표계에 대한 로봇의 위치 및 방향을 나타내 주고 있으며, 아래와 같이 6개의 파라미터 $({}^{W}P_x,\ {}^{W}P_y,\ {}^{W}P_z,\ {}^{W}\alpha,\ {}^{W}\beta,\ {}^{W}\gamma)$ 로 정의한다

$$ {}^{W}T_{R} = Trans({}^{W}P_x, {}^{W}P_y, {}^{W}P_z)\cdot Rot_z({}^{W}\alpha)\cdot Rot_y({}^{W}\beta)\cdot Rot_x({}^{W}\gamma) \qquad (1) $$

(2) 툴 변환(tool transformation) $({}^{6}T_{E})$

용접기 형상을 정의하기 위한 변환 행렬로서 6개의 파라미터 $({}^{E}P_x, {}^{E}P_y, {}^{E}P_z, {}^{E}\alpha, {}^{E}\beta, {}^{E}\gamma)$ 로 정의한다.

$$ {}^{6}T_{E} = Trans({}^{E}P_x, {}^{E}P_y, {}^{E}P_z)\cdot Rot_z({}^{E}\alpha)\cdot Rot_y({}^{E}\beta)\cdot Rot_x({}^{E}\gamma) \qquad (2) $$

(3)로봇 링크 변환(robot link transformation) $({}^{R}T_{6})$

기구학 모델(Kinematic Model)은 로봇의 위치 및 방위 정밀도를 계산하기 위해서 두개의 연속된 좌표계 즉, [i-1]와 [i] 사이에서의 미소 변화를 그 로봇의 링크 파라미터들의 미소 변화들로써 모델링 할 수 있어야 한다. 특히, 두개의 연속된 축이 평행인 경우에 Denavit-Hartenberg Homogenous Transformation 에 y축에 대한 회전 변환 $Rot(y,\beta_i)$ 를 추가 함으로써, 두 연속된 좌표계 사이에서의 위치 및 방위의 미소 변화를 5개의 링크 파라미터들 $[\theta_i,\alpha_i,a_i,d_i,\beta_i]$ 의 미소 변화들로 모델링 할 수 있다. 본 S/W 에서는 캘리브레이션을 고려한 모델을 이용하고 있기 때문에 상기와 같은 5개의 모델 파라미터를 이용하였으며, 여기서, y축에 대한 회전을 고려하는 이유는 두축이 평행한 경우, 작은 형상 오차가 매우 큰 파라미터의 변화를 초래하게 되어 불연속성을 보여주기 때문이다.

$$ {}^{i-1}A_i^m = Rot_z(\theta_i)\ \ Trans(a_i,0,d_i)\ Rot_x(\alpha_i)Rot_y(\beta_i) \qquad (3) $$

(4) 로봇 변환(robot transformation) $({}^{W}T_{E}^{c})$

용접 대상물 기준 좌표계[W]로 부터 툴 좌표계[E] 까지의 위치와 방향을 나타내는 변환행렬은 아래와 같다.

$$ {}^{W}T_{E}^{c} = {}^{W}T_{R}\cdot {}^{R}A_2^m\cdot {}^{2}A_3^m\cdots {}^{n-1}A_n^m\cdot {}^{n}T_{E} \qquad (4) $$

상기 정의된 각 파라미터의 값들은 모두 캘리브레션된 값들을 사용하기 때문에 모든 로봇 시스템은 반드시 캘리브레이션이 수행되어야 하며, 본 S/W 에서는 같은 종류의 로봇 시스템이라 하더라도 상기 파라미터의 값들이 모두 다르기 때문에 각각의 로봇 시스템은 서로 다른 로봇으로 본다.

5. 로봇 캘리브레이션

로봇 캘리브레이션의 목적은 손끝의 위치/방위와 조인트 각 $(\vec{\theta})$ 사이의 보다 정확한 함수 관계를 찾음으로써 오프라인 프로그래밍 응용시 그 절대 위치 정밀도를 향상시키는데 있다. 그 함수 관계는 식(1),(2) 및 (3) 과 같으며, 이 식들을 정의하는 파라미터들의 정밀한 값들을 추정하는 일련의 과정이 캘리브레이션이라 할 수 있다. 본 연구에서의 캘리브레이션 방법은 이미 참고문헌[5] 발표한 바 있으나, 본 논문에서는 이의 알고리즘만을 간단히 요약하려한다. $({}^{W}T_{E}^{c})$ 에서의 로봇 손끝의 위치는 다

음과 같은 간단한 함수 형태로 표현할 수 있다.

$$\vec{F} = \vec{F}(\vec{p}, \vec{\theta}) \tag{5}$$

여기서, $\vec{F} = [f_x, f_y, f_z]^T$는 로봇 손끝의 위치이며, $\vec{p} = [p_1, p_2, p_3, \cdots, p_{np},]^T$ (np : 모델 파라미터의 총 개수)는 모델 파라미터 벡터이며, $\vec{\theta}$ 는 로봇 조인트각 벡터이다. 공칭 기구학 방정식에 사용한 파라미터 값들을 $\vec{p}^N$,실제의 값을 $\vec{p}$,이들의 오차를 $\vec{\varepsilon}$ 라하면, 상기(6)식은 다음과 같이 표현할 수 있다.

$$\vec{F} = \vec{F}(\vec{p}, \vec{\theta}) = \vec{F}(\vec{p}^N + \vec{\varepsilon}, \vec{\theta}) \tag{6}$$

임의의 작업 위치에 해당하는 역기구학 해를 구한 후, 공칭 기구학 방정식을 이용하여 로봇을 움직였다면, 실제 손끝의 위치는 원하는 작업 위치와 미소한 위치 차이가 있을 것이다. 즉,

$$\begin{aligned} \vec{dF} &= \vec{F}^r - \vec{F}(\vec{p}^N, \vec{\theta}) \\ &= \vec{F}(\vec{p}^N + \vec{\varepsilon}, \vec{\theta}) - \vec{F}(\vec{p}^N, \vec{\theta}) \end{aligned} \tag{7}$$

로봇 손끝을 원하는 작업 위치로 정확히 움직이고자 한다면, $\vec{\varepsilon}$ 의 값을 알아야만 한다. 만약 $\vec{\varepsilon}$ 이 작다면 식(7)은 다음과 같이 근사화 할 수 있다.

$$\vec{dF} \cong \left. \frac{\partial \vec{F}}{\partial \vec{\varepsilon}} \right|_{\vec{\varepsilon}=0} \cdot \vec{\varepsilon} \tag{8}$$

여기서 $\left. \frac{\partial \vec{F}}{\partial \vec{\varepsilon}} \right|_{\vec{\varepsilon}=0}$ 은 오차 전파 행렬이다.

일반적으로, 모델 파라미터 값을 추정하기 위해서는 여러 로봇 위치/자세에서 손끝의 위치/자세 ($^W\vec{F}^m$) 및 그때의 로봇 조인트 각 ($\vec{\theta}^m$)을 측정하야한다. 본 연구에서는 이러한 목적을 위해 정밀히 제작된 지그를 제작하였으며, 접촉 센서(touch sensor)를 사용하여 지그의 여러 면을 센싱(sensing)하여 그때마다의 로봇 각 축의 위치와 지그좌표계[W]에 대한 로봇의 손끝의 X,Y,Z 면의 위치를 알 수 있게 된다. 지그의 각 $y-z$ 면에 Mx 번, $x-z$ 면에 My 번, $x-y$면에 Mz 번의 측정을 수행하였을 경우, 기구학 모델을 사용하여 표현된 로봇 손끝의 X,Y,Z 위치와 측정된 위치 사이의 차는 다음과 같이 나타낼 수 있다.

$$(^W dx_E)_i = {}^W f_x(\vec{p}^N + \vec{\varepsilon}, \vec{\theta}_i) - (^W f_x{}^m)_i \tag{9}$$

$$(^W dy_E)_j = {}^W f_y(\vec{p}^N + \vec{\varepsilon}, \vec{\theta}_j) - (^W f_y{}^m)_j \tag{10}$$

$$(^W dz_E)_k = {}^W f_z(\vec{p}^N + \vec{\varepsilon}, \vec{\theta}_k) - (^W f_z{}^m)_k \tag{11}$$

단, $i = 1, 2, 3, \cdots, Mx$, $j = 1, 2, 3, \cdots, My$, $K = 1, 2, 3, \cdots, Mz$ (M : 총 측정 횟수 $= Mx + My + Mz$)

여기서 $^W f_x{}^m$, $^W f_y{}^m$, $^W f_z{}^m$ 은 로봇 손끝이 지그의 $Y-Z$면, $X-Z$면, 또는 $X-Y$면을 접촉하였을 때의 지그 좌표계 [W]에 대하여 표현되어진 로봇 손끝의 위치이다. 또한, 로봇 손끝의 위치 오차를 지그 좌표계 [W]에 대하여 표현할 수 있다.

$$\begin{bmatrix} (^W dx_E)_1 \\ \vdots \\ (^W dx_E)_{Mx} \\ (^W dy_E)_1 \\ \vdots \\ (^W dy_E)_{My} \\ (^W dz_E)_1 \\ \vdots \\ (^W dz_E)_{Mz} \end{bmatrix} = \begin{bmatrix} {}^W w_x(\vec{p}, \vec{\theta}_1^m) & {}^W e_x(\vec{p}, \vec{\theta}_1^m) \\ \vdots & \vdots \\ {}^W w_x(\vec{p}, \vec{\theta}_{Mx}^m) & {}^W e_x(\vec{p}, \vec{\theta}_{Mx}^m) \\ {}^W w_y(\vec{p}, \vec{\theta}_1^m) & {}^W e_y(\vec{p}, \vec{\theta}_1^m) \\ \vdots & \vdots \\ {}^W w_y(\vec{p}, \vec{\theta}_{My}^m) & {}^W e_y(\vec{p}, \vec{\theta}_{My}^m) \\ {}^W w_z(\vec{p}, \vec{\theta}_1^m) & {}^W e_z(\vec{p}, \vec{\theta}_1^m) \\ \vdots & \vdots \\ {}^W w_z(\vec{p}, \vec{\theta}_{Mz}^m) & {}^W e_z(\vec{p}, \vec{\theta}_{Mz}^m) \end{bmatrix} \begin{bmatrix} \vec{\varepsilon}_1 \\ \vec{\varepsilon}_2 \end{bmatrix} \tag{12}$$

이 식(12)을 다음과 같이 간단한 기호로 표현해 보자.

$$\vec{dY} = E\big|_{\vec{\varepsilon}=0} \cdot \vec{\varepsilon} \tag{13}$$

파라미터 오차가 포함된 식(6)의 비선형 기구학 방정식을 직접 이용한다면, 정확한 파라미터 오차 벡터 $\vec{\varepsilon}$ 을 추정할 수 있을 것이다. 이는 아래 식의 함수 값을 최소로 하는 $\vec{\varepsilon}$ 를 구함으로써 얻을 수 있다.

$$\Psi(\vec{\varepsilon}) = [\vec{Y}(\vec{\varepsilon}) - {}^{m}\vec{Y}]^{T} \cdot [\vec{Y}(\vec{\varepsilon}) - {}^{m}\vec{Y}] \quad (14)$$

여기서 $\vec{Y}(\vec{\varepsilon})$는 각 측정 위치에서의 식(6)을 이용하여 계산된 손끝의 단 방향 위치 벡터이며, ${}^{m}\vec{Y}$는 측정된 위치 값들이다. 즉,

$$\vec{Y}(\vec{\varepsilon}) = [{}^{W}f_x(\vec{p}^{N} + \vec{\varepsilon}, \vec{\theta}_1^{m}) \cdots {}^{W}f_x(\vec{p}^{N} + \vec{\varepsilon}, \vec{\theta}_{Mx}^{m}),$$
$${}^{W}f_y(\vec{p}^{N} + \vec{\varepsilon}, \vec{\theta}_1^{m}) \cdots {}^{W}f_y(\vec{p}^{N} + \vec{\varepsilon}, \vec{\theta}_{My}^{m}),$$
$${}^{W}f_z(\vec{p}^{N} + \vec{\varepsilon}, \vec{\theta}_1^{m}) \cdots {}^{W}f_z(\vec{p}^{N} + \vec{\varepsilon}, \vec{\theta}_{Mz}^{m})]^{T} \quad (15)$$

$${}^{m}\vec{Y} = [({}^{W}f_x^{m})_1 \cdots ({}^{W}f_x^{m})_{Mx}, ({}^{W}f_y^{m})_1 \cdots ({}^{W}f_y^{m})_{My},$$
$$({}^{W}f_z^{m})_1 \cdots ({}^{W}f_z^{m})_{Mz}]^{T} \quad (16)$$

이 식(14)는 비선형이므로 $\vec{\varepsilon}$의 최적해는 반복적으로 구할 수 있을 것이다. $\vec{Y}(\vec{\varepsilon} + \Delta\vec{\varepsilon}) = \vec{Y}(\vec{\varepsilon}) + E(\vec{\varepsilon}) \cdot \Delta\vec{\varepsilon}$ 이므로, k 번째 반복 연산에서 식(14)는 다음과 같이 된다.

$$\Psi_k(\vec{\varepsilon}_k) = [\vec{Y}(\vec{\varepsilon}_k) - {}^{m}\vec{Y} + E(\vec{\varepsilon}_k) \cdot \Delta\vec{\varepsilon}_k]^{T} \cdot$$
$$[\vec{Y}(\vec{\varepsilon}_k) - {}^{m}\vec{Y} + E(\vec{\varepsilon}_k) \cdot \Delta\vec{\varepsilon}_k] \quad (17)$$

또한, 이때의 최적해 $\Delta\vec{\varepsilon}_k$는 아래와 같이 구할 수 있다.

$$\Delta\vec{\varepsilon}_k = \{[E^{T}(\vec{\varepsilon}_k)E(\vec{\varepsilon}_k)]^{-1} \cdot E^{T}(\vec{\varepsilon}_k)\} \cdot$$
$$[\vec{\aleph}(\vec{\varepsilon}_k) - {}^{m}\vec{\aleph}] \quad (18)$$

따라서 *(k+1)* 번째의 파라미터 오차 값은 아래와 같다.

$$\vec{\varepsilon}_{k+1} = \vec{\varepsilon}_k + \Delta\vec{\varepsilon}_k \quad (19)$$

상기 과정을 $\Delta\vec{\varepsilon}_k$이 원하는 오차 범위를 만족할 때까지 반복 연산하여 파라미터 오차 값을 추정한다.

6. 용접 대상물 크기변화에 따르는 교시점 위치 /방향 결정

로봇이 작업하는 용접대상물의 종류와 크기가 매번 달라질 경우, 로봇 프로그램을 매번 다시 작성해야하며 이는 로봇 사용을 어렵게 만드는 요인이 된다. 다행히 선체 블록의 경우, 용접 대상물의 종류가 Fig. 2 에서와 같이 한정되어 있기 때문에, 각 종류별로 용접블록의 형상 및 용접선을 정의하는 파라미터들를 정의 할 수 있다. Fig. 6 은 특정 블록 종류를 정의하는 파라미터의 예를 보여주고 있다. 또한, 용접대상 블록의 종류별로 로봇이 교시되어져야 할 교시점 및 작업 시퀀스가 포함된 표준 프로그램(Standard Program)들을 정의할 수 있다. 이 표준 프로그램은 여러 번의 실험을 거쳐 특정 용접대상물에 대하여 이미 완벽하다고 판단된 로봇 프로그램으로서, 같은 종류 내에서 용접선의 위치와 크기가 서로 다른 용접대상물을 위한 프로그램 작성시, 작업 시퀀스 및 용접조건 등을 참조할 수 있는 프로그램이다. 이 표준 프로그램 상의 교시점들은 특정 크기의 용접대상물을 위한 위치 데이터이므로 용접대상물 크기가 달라지면 그 위치데이터는 사용 할 수 없으며, 용접선 위치와 용접조건에 부합되는 새로운 교시 위치를 결정하여야 한다.

이러한 교시 위치(teaching position)들은 용접 대상물의 크기를 결정하는 파라미터(model parameter)의 함수로 정의할 수 있으며, 용접대상물의 종류에 따라 이를 정의하는 파라미터의 종류와 개수도 달라지게 된다. 특정 용접대상물상의 가능한 교시점의 예는 Fig. 6 과 같으며, 본 연구에서는 이러한 교시 점을 H 계열 위치라고 부르겠다. Fig. 7 은 이들의 위치를 정의하는 예를 보여주고 있으며, P_x, P_y, P_z 는 작업 대상물의 벽면으로부터

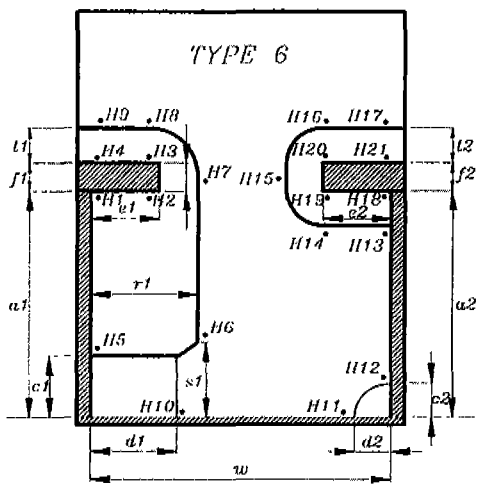

Fig. 6 Typical Parameters defining a section and Feasible teaching points

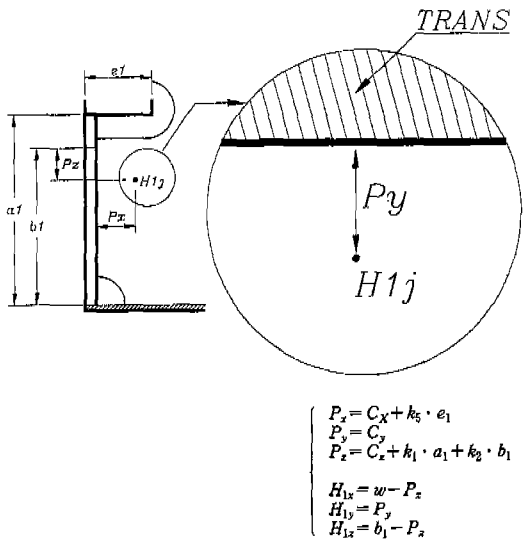

Fig. 7 An example of defining a teaching point position by section geometric parameters

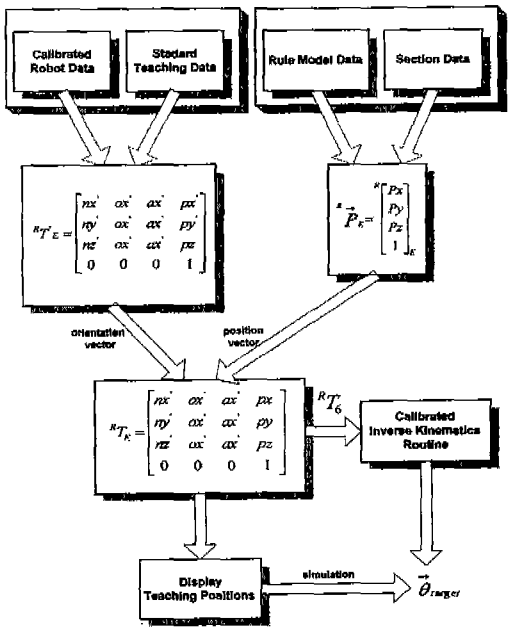

Fig. 8 Schematic flow of determining a teaching point Position/Orientation

의 H 점까지의 거리를 나타내고, H_{1x}, H_{1y}, H_{1z} 는 용접 대상물 좌표계 [W] 에 대한 로봇 손끝위치로서 위에서 정의한 P_x, P_y, P_z 와 용접 대상물 모델 파라미터의 함수로 나타낸 것이다.

교시점의 위치 (H_{ix}, H_{iy}, H_{iz}) 는 상기와 같이 정의할 수 있으며, 교시점의 방향 결정은 용접 조건과 관계가 있으므로 표준 프로그램 상에서의 실험적/경험적인 수치를 이용하여 용접 기울기를 결정할 수 있을 것이다. Fig. 8 은 본 연구에서 임의의 교시점의 위치/방향을 결정하는 과정을 보여주고 있다 . 즉, 용접대상물에 대한 용접기의 위치는 Fig. 7 에서와 같이 용접대상물 크기 파라미터의 함수로서 정의된 룰(Rule) 에 의해 결정하고, 용접기의 기울기는 표준프로그램에서의 교시된 기울기를 이용하게 되는 것이다. 상기와 같은 방법으로 결정된 각 교시점의 위치와 방향이 Fig.8 에서의 ${}^{R}T_{E}$ 이며, 이의 역기구학 해를 구하여 로봇 프로그램을 작성할 수 있다.

7. 캘리브레이션된 역기구학해

로봇 역기구학해는 아래와 같은 비선형 연립 방정식의 해 $(\vec{\theta})$ 이다.

$$t_6(\vec{p}, \vec{\theta}) = {}^{R}T_6 \qquad (20)$$

여기서 $t_6(\vec{p}, \vec{\theta})$ 는 식(2) 를 이용한 순기구학 식이며, $\vec{p}$ 는 전 절에서 추정된 링크 파라미터들의 값, ${}^{R}T_6^{C}$ 는 주어진 로봇 손끝 위치/방향 행렬이다. 일반적으로 로봇의 역기구학해는 아주 간단한 형상을 가진 특별한 로봇의 경우에만 그 닫힌(Closed Form) 해가 존재한다. 하지만, 본 연구에서의 로봇 모델은 4 개의 파라미터를 사용하는 일반적 Denavit-Hartenberg 의 방법을 사용하지 않을 뿐만 아니라, 각 파라미터의 실제값 즉 캘리브레이션를 통해 예측된 값들을 사용하기 때문에 닫힌 해를 구할 수 없다. 따라서 Newton-Raphson 방법과 유사한 수치적 방법을 사용할 수 밖에 없다. 본 연구에서의 역기구학해 방법은 Fig. 9 과 같으며, 자코비안 역행렬을 구할 때 특이 현상이 일어나는 경우 한 개의 조인트 값은 현재 값으로 놓고 나머지 5 개의 조인트 값만을 구하였다.

8. 실행예

본 연구에서 사용된 로봇의 링크파라미터의 공칭값과 4 절에서의 캘리브레이션 방법을 이용하여 얻은 로봇 시스

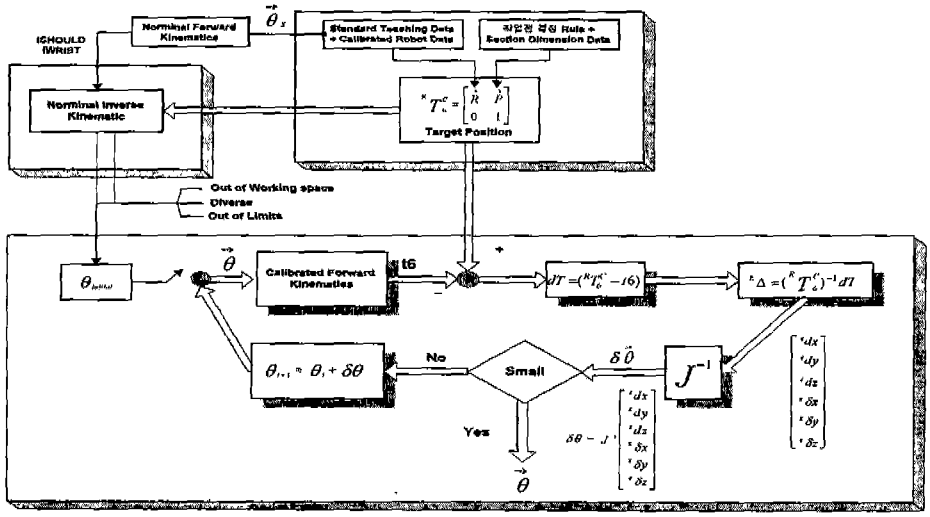

Fig. 9 Schematic flow of solving the calibrated inverse kinematics

Table 1 Model Parameters and their nominal/estimated values

i	Nominal Values of Parameters					Estimated Values After Calibration				
	θ_i	a_i	d_i	α_i	β_i	$\Delta\theta_i$	a_i	d_i	α_i	β_i
1	θ_1	0	0	-90	0	0	0	0	-90.494	0
2	θ_2-90	350	0	0	0	0.482	349.515	0	0.737	-0.540
3	θ_3	0	80	-90	0	-0.248	0	80	-89.474	0
4	θ_4	0	350	90	0	-0.341	2.549	350.911	89.474	0
5	θ_5	0	-80	-90	0	0.075	0	-60	-89.951	0
6	θ_6	0	0	0	0	0	0	0	0	0

	P_x	P_y	P_z	α	β	γ	P_x	P_y	P_z	α	β	γ
W-R	430	810	340	-90	0	0	438.174	814.432	340.116	-90.368	-1.256	0.279
E-E	0	0	410	0	-18	0	0.287	2.941	408.771	0	18	0

(Units are mm or degrees)

Table 2 Examples of Calibrated inverse kinematic Solutions

[Example 1] Exact Solution : θ_1=10, θ_2=-20, θ_3=30, θ_4=-40, θ_5=50, θ_6=-60						[Example 2] Exact Solution : θ_1=10, θ_2=-20, θ_3=30, θ_4=-40, θ_5=0, θ_6=-60						
	Iteration No.					Iteration No.						
	1	2	3	4	5	1	2	3	4	5	6	7
θ_1	20.0000	11.4339	10.1651	9.9950	10.0000	20.0000	11.7740	11.1509	10.2582	10.0097	10.0000	10.0000
θ_2	-10.0000	-22.1877	-20.0336	-19.9987	-20.0000	-10.0000	-22.5744	-20.0442	-19.9457	-19.9978	-19.9999	-20.0000
θ_3	40.0000	28.3625	30.0656	29.9987	30.0000	40.0000	24.8244	29.2062	29.8128	29.9948	29.9999	30.0000
θ_4	-30.0000	-45.5695	-40.2567	-40.0051	-40.0000	-30.0000	-32.0406	-39.5912	-38.6735	-39.9588	-39.9999	-40.0000
θ_5	60.0000	58.5942	50.3315	49.9977	50.0000	10.0000	11.6456	1.5662	0.2524	0.0083	0.0000	-0.0000
θ_6	-50.0000	-59.0976	-59.2504	-59.9976	-60.0000	-50.0000	-71.4780	-65.9375	-61.2809	-60.0394	-60.0000	-60.0000

(Units are mm or degrees)

템의 모델 파라미터는 Table 1 와 같다. 이 데이터는 컴퓨터 상에 로봇을 정확히 모델하는 데 사용되어지며, Fig. 10 은 이 데이터를 이용하여 자동 생성된 로봇 시스템의 예이다. 또한 Table 2 는 6 절에서의 캘리브레이션된 역기구학해 방법을 이용하여 역기구학해를 구한 예 이며, 퇴보현상이 일어나리라 기대했던 ($\theta_5 = 0$) 근처에서도 정밀한 역기구학 해를 구할 수 있었다.

본 S/W 에서는 한 개의 선체 블록에 있는 각각의 용접 대상물에 대하여, 사용할 로봇 시스템(캘리브레이션 데이터) 및 표준 프로그램을 선택하게 하면, 이들을 이용하여 빠른 시간 내에 새로운 교시점 데이터(캘리브레이션된 역기구학해)를 자동 생성하도록 하였다. Table 3 은 본 오프라인 시스템이 자동 생성한 작업 프로그램이며, Fig.11 는 이 작업 프로그램을 실제 로봇에 입력하여 용접을 수행하는 모습을 보여주고 있다. 용접 결과는 수직용접, 수평용접 및 코너링 용접 모두 만족할 만 하였으며, 현재 조선 산업 현장에서 성공적으로 실험 가동 중이다.

9. 결 론

로봇을 사용할 때 작업 대상물의 크기가 매번 혹은 자주 변화하는 경우, 오프라인 프로그램밍 방법은 실제 상황에서 실제 로봇을 이용하지 아니하고 컴퓨터에서 프로그램밍을 한다는 측면에서 효과적일 수 있다. 일반적인 오프라인 S/W 에서는 컴퓨터 상에서 모델된 작업 대상물

Table 3 An example of robot job program generated by Offline Programming System

```
001 HOME  V=20.0 (%) HF=1
002 HOME  V=20.0 (%) HF=2
003 MOVJ  T00 V=20.0 (%) PL=0
004 MOVL  T01 V50.0  PL0 D0
005 MOVL  T02 V50.0  PL0 D0
006 TOUCH V=10.0  X0  Y1  Z0  L100 P30
007 IMOV  V=30.0  x=0    y=-10  z=0
008 TOUCH V=10.0  X1  Y0  Z0  L100 P31
009 IMOV  V=30.0  x=-10  y=0    z=0
010 IMOV  V=30.0  x=7    y=7    z=0
011 GETP  P01
012 IMOV  V=30.0  x=-7   y=-7   z=0
013 MOVL  T03 V50.0  PL0 D0
...
... ( 중략)
...

129 WEAVE P18 P09 WD0    AD0    WVF1
130 WSET  MAIN=12  SENSOR=0
131 MOVC  P07 P08 V=20.0
132 WSET  MAIN=11  SENSOR=0
133 MOVL  P05 V30.0  PL0 D0
134 ARCOF END=2
135 MOVL  T27 V50.0  PL0 D0
136 MOVL  T28 V50.0  PL0 D0
137 MOVL  T29 V50.0  PL0 D0
138 HOME  V=20.0 (%) HF=2
139 HOME  V=20.0 (%) HF=1
;

;
T00 = -0.0349 -0.4520  0.8396 -0.0400 -0.9120  0.0432
T01 = -0.1275  0.2631  0.5408 -1.2468 -1.3922 -0.7750
T02 =  0.0833  0.5146  0.2081 -1.5167 -1.1155 -1.4545
T03 =  0.1900  1.2658  0.0096 -1.2339 -0.8350 -1.8697
T04 =  0.1895  1.3174 -0.0310 -1.2818 -0.9119 -0.6301
T05 =  0.1671  1.4022 -0.0232 -1.1608 -1.0478 -1.4440
...
... ( 중략)
...
T19 = -0.2421  1.5152  0.3701  0.3867 -1.6159  0.2368
T20 =  0.1996  1.1355  0.6742  0.8527 -1.4861  0.3832
T21 =  0.1222  0.4841  0.1888  1.6677 -1.2957  1.4040
;
```

과 모델된 로봇을 이용하여, 온라인 프로그램과 똑같은 과정을 거쳐 프로그램밍 작업을 수행하게 된다. 따라서, 일반적인 오프라인 프로그램밍 역시 매우 많은 시간이 걸리며, 일반적으로 온 라인의 경우 보다 더 많은 시간이 소요되는 것이 보통이다. 게다가, 오프라인 프로그램밍은 온라인 프로그램밍과는 달리 로봇과 작업대상물과의 위치관계, 로봇 시스템의 정확한 함수관계 등을 알기 위한 캘리브레이션을 수행해야하는 번거러움이 있다. 따라서, 오프라인 프로그램 방법이 온라인 프로그램 방법에 비해

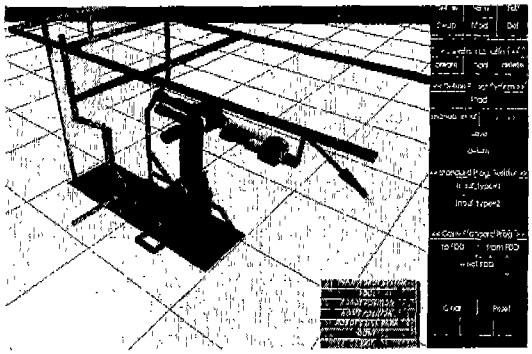

Fig.10 A robot system generated using Calibrated parameters in the Offline Programming system

Fig.11 View of a robot performing a welding task by Program generated through the Offline Programming System

많은 장점을 갖기 위해서는 첫째로 로봇 프로그램을 작성하는 데 있어서 시간이 많이 들지 않도록 오프라인 프로그램밍 자동화가 이루어져야 하며, 둘째로는 온 라인 프로그램과 같은 정밀도를 손쉽게 확보할 수 있는 효과적인 캘리브레이션 방법이 함께 갖추어져야 하는 것이다.

본 연구에서는 첫째로 오포라인 프로그래밍에서 문제가 되는 로봇의 절대 위치 정밀도를 높이기 위해 아아크용접기를 장착한 로봇을 위한 효과적인 캘리브레이션방법[5]을 개발하여 오프라인 프로그램밍의 신뢰성을 높일 수 있도록 하였다. 이러한 캘리브레이션 실험을 통해 얻어진 기하학적 파라미터들을 이용하여 컴퓨터상의 로봇 시스템과 작업 대상물이 실제 작업현장의 조건과 거의 일치할 수 있도록 하였다. 둘째로는, 작업 대상물의 크기는 달라도 그 형태가 같은 경우에는 형태 별 작업점 결정 규칙과 모든 작업 조건을 만족하며 성공적으로 수행되어진 증명된 표준 작업 프로그램을 참조하여 새로운 작업 프로그램

을 생성하도록 함으로써, 경험적인 요소를 가미한 프로그램 생성을 빠른 시간 내에 수행할 수 있도록 하였다. 즉, 일반적인 오프라인 프로그램에서는 사용자가 매번 작업점을 결정해 주던 것을 일종의 작업점 결정 규칙에 의거하여 결정해 주도록 하였고, 이 작업점 결정 규칙은 사용자의 경험과 능력에 따라 변화할 수 있도록 하였다.

본 연구는 특정 응용을 위한 오프라인 프로그램밍 자동화를 위한 것이지만 그 연구 결과는 다른 산업 분야에도 적용될 수 있으리라 믿는다.

10. 참고 문헌

1. R.O.Buchal, D.B.Cherchas, J.P.Dun can, "Simulated Off-Line Programming of Welding Robot", The International Journal of Robotics Research, Vol.8,No.3, pp.31-43, 1989.
2. TatsuoMIYAZAKI,YoshioNAKASHIMA, 平成5年, "ROBOT APPLICATION TECHNOLOGY FOR SHIYARD CIMS IMPLEMENTATION", 西部造船會 第86回, pp.199-211.
3. Deneb Robotics, Inc., "IGRIP User's Manual", 1993.
4. Technomatic, Inc., "Robcad User's Manual", 1994.
5. 범 진환, "아아크 용접 로봇의 오프라인 프로그램 응용을 위한 효과적 캘리브레이션 방법 연구", 한국 정밀 공학회, 제13권,제1호,pp. 131-142, 1996.1.
6. Martin Wihsbeck, "Activities of IGM in Ship-Industry : Project, Experiences and Developments", IGM Robotersysteme AG, Austria, 1992.
7. C.H.Menq and J.H.Borm, "Statistical Measure and Characterization of Robot Errors", IEEE International conference on Robotics and Automation, Philadelpia,26-28, 1988.
8. 범진환, 임생기, 손명현, "손목 오프셋을 갖는 6축 로봇를 위한 효과적인 역기구학 해 방법", 대한 기계학회 논문집 제18권 제6호, pp. 1421-1429, 1994.
9. P. Howie, "Graphic Simulation for Offline Programming," Robotics Today, pp.63-66, 1984.
10. G. Duelen, H. Stahlmann, and X.Liu, "An Off-line Planning and Simulation System of Coating Robots", Annals for CIRP, vol. 38, no.1,pp.369-372, 1989.
11. Denavit,J and Hartenberg, R.S., "A Kinematic notation for lower pair mechanism based upon matrices", ASME Journal of Applied Mechanics, vol. 22, pp.215-221. 1955.
12. Paul, R. P., "Robot Manipulator", MIT Press, Cambridge, MA, 1981.