

OLP 기반 산업용 로봇 시뮬레이션을 위한 자동화 프로그램 모듈 개발

이수준*, 이세한⁺, 박종근⁺⁺

(논문접수일 2008. 5. 7, 심사완료일 2008. 11. 12)

Development of Automation Program Module for OLP based Industrial Robot Simulation

Soo Jun Lee*, Se-Han Lee⁺, Jong-keun Park⁺⁺

Abstract

Interactive Graphic Robotics Integrated Programming(IGRIP) can handle various types of robot models and can exchange graphic or numerical data easily with other CAD software. In a cutting process of shape-steels, however, IGRIP is inconvenient because the users must generate all the tag points manually. In this study we developed an automation program module in order to generate the tag points automatically in IGRIP. This program can read and analyze the macro data containing the information for cutting processes of shape-steels and can generate automatically the parts, the devices, the tag points and the Graphic Simulation Language(GSL) program files useful in IGRIP.

Key Words : Gantry-type robot(갠트리형 로봇), Shape-steel cutting(형강 절단), Automation Program Module(자동화 프로그램 모듈), GSL(그래픽 시뮬레이션 언어), Tag point(교시점)

1. 서론

선박 건조 공정은 크게 소재의 절단과 같은 가공 공정, 소재의 용접으로 대표되는 조립 공정 그리고 블록의 연결 공정 등의 순서로 나눌 수 있다.

절단 및 용접과 같은 가공 및 조립공정은 단위 작업 별로

자동화 설비가 충분히 갖추어져 있으나 공정개선을 위한 전체 자동화 체계는 아직까지 일부 현장에서만 개발 및 실용화가 진행 중에 있다.

최근에는 신규로 제작되는 선박에 대하여 이중선체 구조가 적용되고 있으므로 선박 재료에서 형강부재가 차지하는 비중이 증가되고 따라서 형강 절단 자동화에 대한 기술개발

* 경남대학교 대학원 기계공학과

+ 경남대학교 기계자동화공학부

++ 교신저자, 경남대학교 기계자동화공학부 (jkpark@kyungnam.ac.kr)

주소: 631-701 경남 마산시 월영동 449

의 필요성이 증대되고 있다.

형강 절단 자동화에 관한 기술 중에서 로봇 제작 및 제어 분야에서는 실용화 연구가 많이 진행되어 있다. 그러나 다축 로봇을 사용한 절단 공정의 자동화에 관한 연구는 아직도 낙후되어 있어서 이에 대한 연구개발이 시급하다⁽¹⁾.

로봇에 의한 자동화 체계는 대부분 온라인 방식으로 수행되고 있다. 이러한 온라인 방식은 작업경험이 풍부한 전문가에 의해서 교시, 성능 실험이 이루어져야 한다.

또한, 복잡한 작업의 경우에는 교시에 많은 시간이 소요되고 시스템의 일부분의 변경에 따른 성능 평가과정에서도 많은 시간과 비용이 소요되는 문제점이 있다⁽²⁾.

이런 문제점은 실제 작업과 동일한 과정을 가진 시뮬레이션(simulation)인 OLP(off-line programming) 체계를 이용함으로써 해결될 수 있다. 이런 OLP 체계에서는 3차원 그래픽을 이용하여 가상공간에서 작업 환경과 로봇을 구현한 후 교시에서 성능 평가까지의 일련의 과정을 시뮬레이션 할 수 있기 때문에 기존의 온라인 프로그래밍의 단점을 보완할 수 있다⁽³⁾.

현재 상용화 되어 있는 OLP 체계에는 Cimstation, WORKSPACE, GRASP, ROBCAD 그리고 IGRIP등이 있다. 국내에 개발된 OLP로서 Jeong⁽⁴⁾, Kim⁽⁵⁾ 그리고 Lee⁽⁶⁾등에 의해 개발된 OLP가 있으며 Cha⁽⁷⁾, Chung⁽⁸⁾등의 IGRIP을 기반으로 한 시뮬레이션에 대한 연구가 있다.

IGRIP(Interactive Graphic Robotics Integrated Programming)은 여러 가지 장점이 있지만 그 중에서도 특히 인터페이스 작업이 용이하고 다양한 소프트웨어와 데이터 교환이 가능하다. 하지만 로봇이 절단 작업을 수행하도록 하기 위하여 로봇 조작자가 블록 도면을 이용하여 수동으로 직접 교시점(tag point)을 생성해야 하는 불편이 따른다⁽⁸⁾.

본 연구에서는 IGRIP을 이용하여 로봇의 절단작업에 관한 시뮬레이션을 수행할 때 지금까지 수동으로 생성 시켰던 교시점을 자동으로 추출하는 자동화 프로그램 모듈(Automation Program Module)을 Microsoft Visual C++ 6.0을 이용하여 개발하였다.

좀 더 구체적으로 언급하면, 형강 절단을 위한 작업 정보가 들어있는 macro 데이터를 자동화 프로그램 모듈을 사용하여 판독 및 분석한 후 IGRIP 상에서 사용할 수 있는 부품(part), 장치(device), 교시점 그리고 시뮬레이션을 위한 GSL(Graphic Simulation Language) 프로그램을 자동으로 생성시켰다.

2. 시뮬레이션 장치의 구성

본 연구는 형강 절단 자동화에 관한 것으로서 전체 자동화 체계 중 갠트리형 로봇을 이용한 형강 절단 작업에 초점을 맞추고 있으며 이에 대한 시뮬레이션은 Table 1과 같은 단계로 수행된다.

Table 1의 단계를 구체적으로 설명하면 다음과 같다.

1. 갠트리형 로봇의 각 부품을 CATIA에서 모델링 한 후 IGRIP에서 조립하여 장치 파일을 만든 후 판독 및 기구학적 요소 등의 운동 특성을 부여한다.

2. 형강 절단을 위한 작업 정보들이 들어있는 macro 데이터를 자동화 프로그램 모듈을 이용하여 판독 및 분석한 후 수치데이터들을 추출한다.

3. 추출된 수치 데이터들 중 일부는 형강의 형상을 정의하는데 사용된다. 이는 IGRIP 환경에서 사용할 수 있는 부품 및 장치 파일 형식으로 생성되어 작업자가 원하는 경로에 자동으로 저장된다.

4. 추출된 수치 데이터들 중 또 다른 일부는 교시점을 생성하는데 사용된다. 이 교시점 데이터는 작업자가 원하는 경로에 일정한 배열을 지닌 text 파일 형식으로 저장된다.

5. 로봇 이동과 형강 이동을 위한 GSL 프로그램이 자동으로 생성되어 작업자가 원하는 경로에 저장된다. 여기서 형강

Table 1 Procedures of simulation

수행 단계	내 용
로봇 형상 정의	<ul style="list-style-type: none"> • CATIA를 이용한 모델링 • IGRIP에서 조립 및 운동특성 부여
Macro 데이터분석	<ul style="list-style-type: none"> • 자동화 프로그램 모듈을 이용한 XML 분석 • 자동화 프로그램 모듈을 이용한 수치데이터 추출
형강 형상 정의	<ul style="list-style-type: none"> • 부품 File 정의(FileName_PRT) • 장치 File 정의(FileName_DEV)
교시점 생성	<ul style="list-style-type: none"> • 교시점 자동 생성(tag name, x, y, z, yaw, pitch, roll)
GSL 프로그램	<ul style="list-style-type: none"> • 로봇 이동 프로그램(FileName_ROBOT.gsl) • 교시점 저장 프로그램(FileName_ROBOT.gsl) • 형강이동 프로그램(FileName_SHAPE.gsl)
시뮬레이션	<ul style="list-style-type: none"> • IGRIP을 기본으로 한 사용자 환경 • 자동화 프로그램 모듈을 이용한 형강 절단 과정

이동을 위한 GSL 프로그램에는 앞에서 생성한 교시점을 저장하는 기능이 포함된다.

6. 형강절단 시물레이션을 하기위한 기본적인 사용자 환경으로 IGRIP이 사용되고 자동화 프로그램 모듈을 통하여 앞에서 언급한 데이터들이 추출된 후 시물레이션이 수행된다.

3. 로봇 형상의 정의

IGRIP에서는 부품들이 모여 독립적인 움직임을 갖는 장치가 만들어지고 다시 장치들이 모여 작업장(workcell)이 구성되는데⁽⁹⁾, 본 연구에서는 갠트리형 로봇과 형강이라는 2개의 장치를 가지고 작업장을 구성하였다.

CATIA 에서 모델링 한 각 부품을 IGRIP용 3D 모델로 변환하였다. 이를 위해서 CATIA의 모델을 부품 단위로 나누어서 STL 파일로 변환하고 IGRIP에서 불러들인 후 다시 조립하는 과정을 거쳤다. CATIA와 IGRIP 환경 하에서 갠트리형 로봇의 모델을 Fig. 1에 나타낸다.

4. Macro 데이터의 분석

4.1 XML의 수치데이터 추출

XML은 web 환경에서 구조화된 문서와 데이터를 교환하기 위한 표준 포맷이다. 단순한 텍스트 포맷(flat file format)을 사용하지 않고 XML을 사용하는 이유는 부가적인 정보 없이 의미론적(semantic)으로 구조화된 데이터를 표현할 수 있기 때문이다⁽¹⁰⁾.

로봇이 형강을 각 부재 별로 절단할 수 있도록 설계 프로그램으로부터 XML 형식으로 된 macro 데이터를 제공 받게

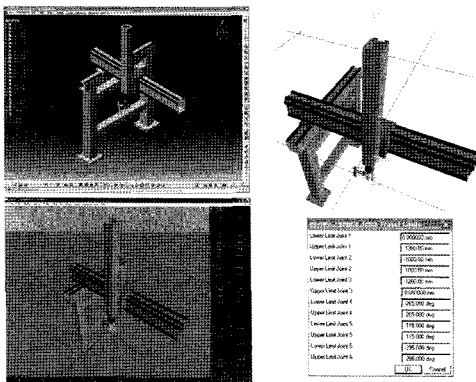


Fig. 1 Gantry-type robot model

된다⁽¹¹⁾. XML을 대상 응용 프로그램에 적용하기 위해서는 XML의 정보에 대한 정확한 판독 및 분석이 중요하다. 형강절단 정보를 담은 XML의 도해적 구조를 Fig. 2에 나타낸다.

XML의 도해적 구조에서 중요한 정보는 형강 절단 명칭

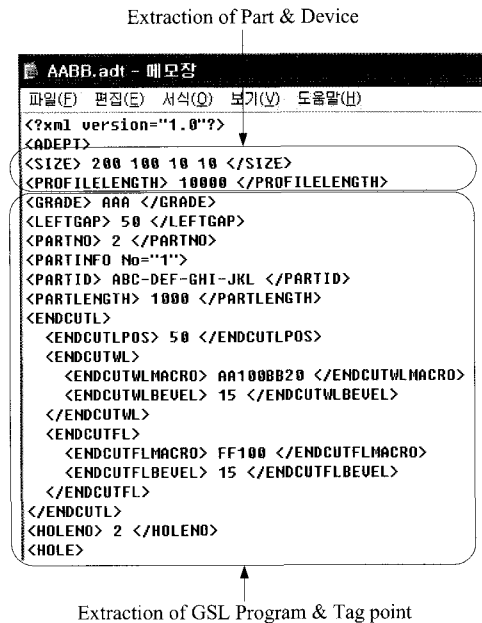


Fig. 2 XML data structure for shape-steel cutting

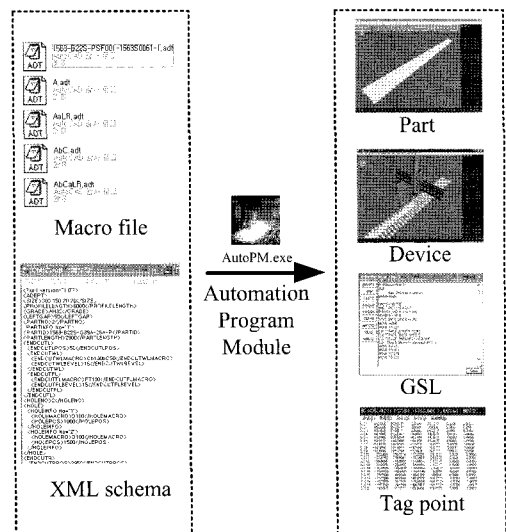


Fig. 3 Procedure of extracting the data from XML in Macro file

뒤에 표기된 수치 데이터들이다. 이 수치 데이터를 이용하여 형강의 모양과 크기 그리고 절단 경로의 종류와 방향을 결정한다.

macro 데이터로부터 수치 데이터를 추출하기 위해서 자동화 프로그램 모듈을 이용한다. 추출된 데이터는 IGRIP용 부품, 장치, GSL 프로그램 그리고 text 파일 형식으로 된 교시점을 만드는데 사용된다. Fig. 3은 자동화 프로그램 모듈을 통해서 이루어지는 데이터 추출 과정을 나타낸다.

4.2 형강 형상의 정의

일반적으로 CAD/CAE/CAM 프로그램은 각 프로그램마다 고유의 파일 형식을 지니고 있기 때문에 같은 형상의 데이터라 할지라도 다른 프로그램에서 이것을 직접 사용할 수 없다. 따라서 데이터 호환에 따른 문제를 해결하기 위하여 공통으로 사용할 수 있는 중립적인 파일이 필요하며 그 종류에는 DXF, STEP, IGES 등이 있다. IGRIP에서는 CPU의 부하를 줄이기 위하여 표면 정보만을 가진 파일을 사용하므로 파일 형식이 비교적 간단하다.

IGRIP용 부품 파일의 구조를 생성하기 위해서 먼저, 형강의 형상을 나타내는 각각의 수치를 변수로 정의한다. 다음으로, XML에서 형강의 형상을 나타내는 수치 데이터를 추출, 분석한 후 앞서 정의된 변수에 적용하여 IGRIP용 부품 파일을 생성한다. Table 2는 형강의 치수를 정의하는 변수에 대한 설명이고 Fig. 4는 형강의 치수를 정의하는 변수를 나타낸다. Fig. 5는 자동화 프로그램 모듈을 이용하여 생성한 형강의 부품 파일의 예를 나타낸다.

자동으로 생성된 부품 파일은 물체의 기하학적인 정보와 그 물체의 전체 공간상에서의 좌표를 데이터로 담고 있다. 이러한 물체들이 여러 개 또는 단독으로 하나의 부품 조합을 구성하게 되는데 이를 장치(device)라고 부른다. 부품 파일은 단순한 형상에 불과하지만 장치 파일은 기구학적 특성을 표현하는 정보와 이 장치를 구성하는 부품에 대한 정보 및 부품 사이의 상호 관계를 담고 있다⁽¹²⁾.

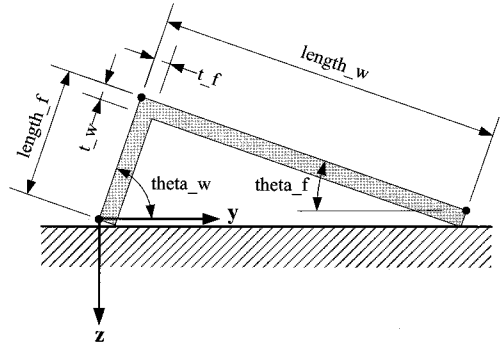
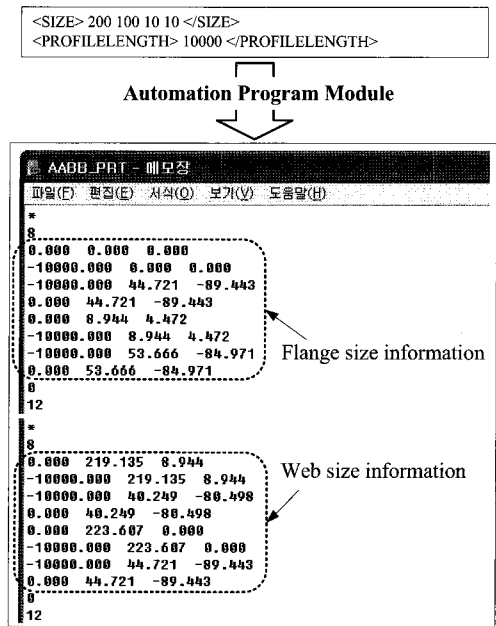


Fig. 4 Variable definition of the shape-steel dimension



Load to IGRIP

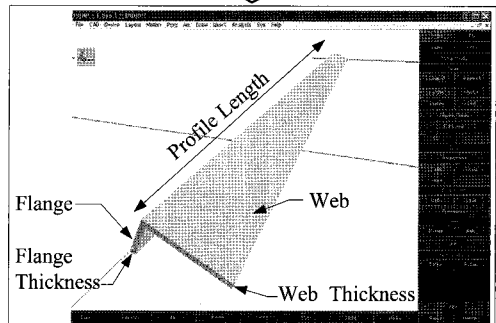


Fig. 5 Part files of the shape-steel

Table 2 Explanation of variable definition

Variable	Explanation
length_w	Web Width
length_f	Flange Width
t_w	Web Thickness
t_f	Flange Thickness
theta_w	Web Angle
theta_f	Flange Angle

본 연구에서는 IGRIP용 장치 파일의 형식 구조를 파악한 후 자동화 프로그램 모듈을 이용하여 부품 및 장치 파일이 자동으로 생성되도록 하였다.

4.3 교시점의 생성

교시점은 3차원 공간상에서 한 강체의 위치(position)와 자세(orientation)를 나타내는 6개의 수치 데이터로서 IGRIP에서 형강 절단 경로를 계획하기 위하여 사용된다.

즉, 교시점은 로봇의 말단효과장치(end effector)가 움직여야 하는 위치를 의미하는 좌표축 데이터와 말단효과장치의 접근 방향(자세)을 의미하는 접근 데이터로 구성된다.

Fig. 6은 교시점의 정의를 나타낸다. macro 파일이 열릴

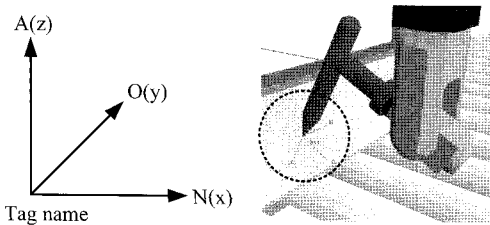


Fig. 6 Definition of tag point

Table 3 A data structure of a tag point

Tag point name	Coordinate axis data			Approach data		
	x	y	z	yaw	pitch	roll
LC1	-89.0	336.0	0.316	26.5	0.0	0.0

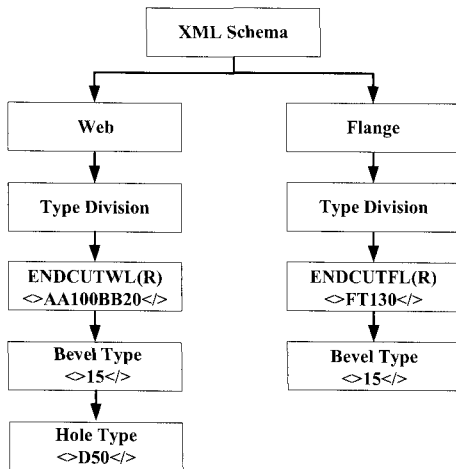


Fig. 7 Procedure of XML analysis for tag points

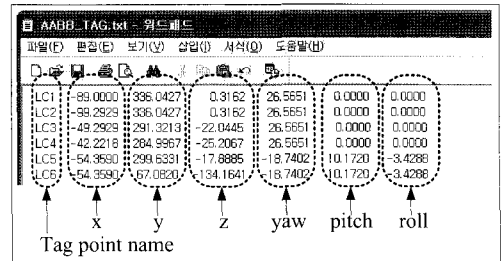
때 형강 절단 경로의 형태에 따라 이 좌표축 데이터와 접근 데이터가 자동화 프로그램 모듈을 통해 자동으로 생성되도록 하였다. Table 3은 이 교시점의 예를 보여준다.

교시점을 생성하기 위한 데이터를 XML로부터 추출하기 위해서는 절단 경로에 대한 정보의 구조를 파악해야 한다. Fig. 7에서는 교시점을 생성하기 위해서 XML로부터 수치데이터를 추출하는 순서도를 보여주고, Fig. 8에서는 자동화 프로그램 모듈을 이용하여 생성된 교시점을 IGRIP용 장치 파일에 저장하는 과정을 설명한다. 이렇게 생성된 교시점은 text 파일 형태로 지정된 장소에 저장된다.

4.4 GSL 프로그램의 생성

GSL 프로그램은 장치 또는 작업장을 시뮬레이션 하기 위한 컴퓨터 언어로서 일반적인 OLP 체계에서 사용되는 로봇 관련 프로그램이며 각각의 장치마다 GSL 프로그램을 저장해야만 시뮬레이션이 가능하다.

형강 절단에 관한 시뮬레이션을 구현하기 위해서 필요한 GSL 프로그램은 형강 이동 및 교시점 저장 기능을 담당하는 부분, 로봇 이동을 담당하는 부분 그리고 형강과 로봇의 입출력을 담당하는 부분 등 3가지로 구성된다. Table 4는 이와 같은 GSL 프로그램의 구성을 나타낸다.



Load to IGRIP

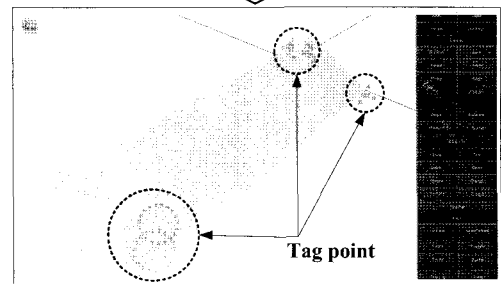


Fig. 8 Tag point generated in Automation Program Module

형강과 로봇의 입출력 신호는 형강 이동 및 교시점 저장 담당 GSL 프로그램과 로봇 이동 담당 GSL 프로그램에 모두 포함되어야 하므로 결과적으로 두 개의 장치에 두 개의 GSL 프로그램이 필요하게 된다.

자동화 프로그램 모듈을 통해서 앞서 언급한 두 개의

Table 4 GSL program for shape-steel cutting

GSL 프로그램	내용	파일명
형강 이동 및 교시점 저장	형강을 작업장으로 이동시키고 교시점에 관한 text 파일을 불러 들여 형강에 위치시킨다.	SHAPE.gsl
로봇 이동	로봇의 절단 작업을 지시한다.	ROBOT.gsl
형강과 로봇 입출력	형강과 로봇의 입출력 신호를 담당한다.	직접 입력

```

GSL generation function in Automation Program Module
Microsoft Visual C++ - [lgripDoc.cpp -]
Edit View Insert Project Build Tools Window Help
// GSL_ROBOT 파일 생성 *****
gsl_data1 = "PROGRAM ROBOT";           gsl_data1 +=
gsl_data1 += "VAR";                   gsl_data1 +=
gsl_data1 += "----- Main Declaration Section";  gsl_data1 +=
gsl_data1 += "BEGIN MAIN";           gsl_data1 +=
gsl_data1 += "-----";               gsl_data1 +=
gsl_data1 += "UNITS= METRIC";         gsl_data1 +=
gsl_data1 += "$MOTYPE= STRAIGHT";     gsl_data1 +=
gsl_data1 += "$SPEED_MODE = ACTUAL";  gsl_data1 +=
gsl_data1 += "$SPEED = 1000";        gsl_data1 +=
gsl_data1 += "-----";               gsl_data1 +=
gsl_data1 += "WAIT UNTIL DIN[0] == ON";  gsl_data1 +=
gsl_data1 += "MOVE HOME";            gsl_data1 +=
    
```

AABB ROBOT.gsl

```

AABB_ROBOT.gsl - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
PROGRAM ROBOT
VAR
----- Main Declaration Section
BEGIN MAIN
UNITS= METRIC
$MOTYPE= STRAIGHT
$SPEED_MODE = ACTUAL
$SPEED = 1000
WAIT UNTIL DIN[0] == ON
MOVE HOME
MOVE TO LC1
CLK ("TRACE LENGTH 1024")
CLK ("TRACE COLOR $RED")
CLK ("TRACE THICKNESS MEDIUM")
CLK ("TRACE ON PART_DEVICE:AABB_PRT")
MOVE TO LC2
MOVE TO LC3
MOVE TO LC4
MOVE TO LC5
MOVE TO LC6
MOVE TO LC7
MOVE TO LC8
MOVE TO LC9
MOVE TO LC10
MOVE TO LC11
MOVE TO LC12
CLK ("TRACE OFF")
MOVE HOME
DOUT[0] = ON
WAIT UNTIL DIN[0] == OFF
DOUT[0] = OFF
WAIT UNTIL DIN[0] == ON
MOVE TO RT
    
```

Fig. 9 Example of GSL program generation

GSL 프로그램은 자동 생성되며 지정된 경로에 자동으로 저장된다. Fig. 9는 GSL 프로그램 생성 과정의 예를 나타낸다.

5. 절단경로의 도출

5.1 형강의 기본 좌표

교시점 데이터는 위치정보 3개에 자세정보 3개를 더하여 총 6개로 구성되고 이것을 Fig. 10에 나타낸다. 이것은 앞에서 언급한 x, y, z, yaw, pitch, roll에 해당한다.

여기서, yaw는 로봇의 토치가 형강 부재 경사면과 수직이 되도록 할 때, pitch는 절단면에 bevel이 있는 경우일 때 그리고 roll은 토치의 진행 방향을 결정할 때 사용한다. 여기서 형강은 x축 방향으로 이동한다.

5.2 경로 계산 알고리즘

형강의 절단 경로를 계산하기 위하여 추출된 XML의 수

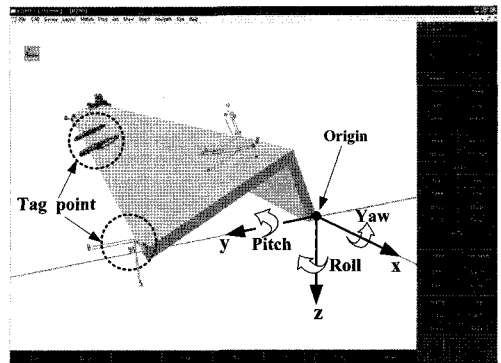


Fig. 10 Basic coordinates of the shape-steel

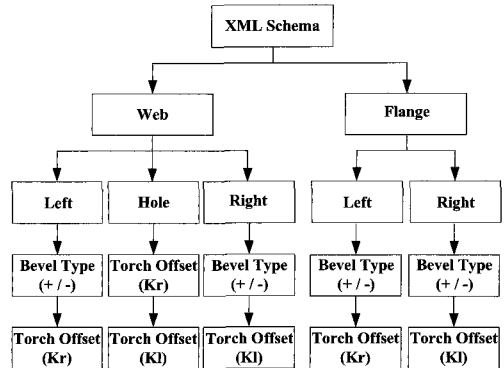


Fig. 11 Flow chart of path generation for shape-steel cutting

치 데이터를 이용하면 절단 경로의 형태를 그 종류별로 분류할 수 있다. XML 형식으로 표시되어 있는 데이터들은 일련의 순서를 지니고 있다. 이 데이터들의 순서를 고려하여 절단 경로를 계산할 수 있는 알고리즘을 만든다. 이것을 Fig. 11에 표시하였다.

5.3 형강 절단 형태의 분류

형강 절단을 위해서는 형강은 그 형상과 길이에 따라 여러 개의 절단 구간들로 나누어진다. 나누어진 구간 내에서 형강은 다시 그 구간 별로 다양한 형태로 절단 된다. 형강이 절단 되는 형태를 분류하여 문자와 수치로 표현할 수 있으며, 이런 문자와 수치를 조합하면 다양하고 복잡한 형태의 절단 경로를 도출할 수 있다.

이러한 일련의 과정을 자동화하기 위해서 가장 기본적인 절단 형태를 분류하여 문자와 수치로 함수 화하고 이를 다시 조합하여 요구되는 절단 경로를 도출한다. 이런 작업은 교시

점의 x, y, z값과 yaw, pitch, roll을 계산하기 위한 필수 과정으로 이보다 앞서 수행되어야 한다. Fig. 12는 'L' 형 형강의 전개도에 표시된 절단 경로의 예를 나타낸다. 여기서 'AA'와 'BB'는 절단 경로의 형태를 나타내는 문자이다.

Table 5는 경로 도출을 위해서 macro 데이터를 형태별로 분류한 것이다. 여기서, Type 1은 web 절단에 대한 기본 변수이고 Type 2는 Type 1의 조합을 나타낸다. Type 3은 flange 절단에 대한 기본 변수이고 Type 4는 Type 3의 조합을 나타낸다. Type 5는 구멍 형태의 절단을 나타내며 web 절단에서만 사용된다.

'AABB' 함수에 대한 절단 순서의 예를 Fig. 13에 나타내며 구체적인 설명은 다음과 같다.

1. macro 데이터에서 'AA100BB50'을 추출한다.
2. AA를 Type 1에 보내고, '100'을 변수 yy에 저장한다.
3. BB를 Type 1에 보내고, '50'을 변수 nn에 저장한다.
4. Type 1을 조합시켜 Type 2를 만든다.
5. 자동화 프로그램 모듈 안에 내장되어 있는 사용자 함수

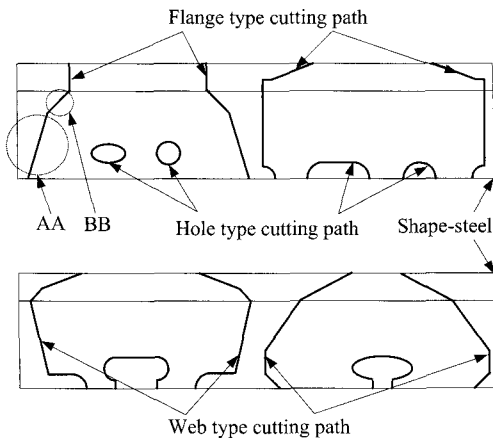


Fig. 12 Example of cutting path type

Table 5 Types of cutting path in macro data

Web		Flange		Hole
Type 1	Type 2	Type 3	Type 4	Type 5
A	AA	FA	FAA	HA
B	AB	FB	FAB	HB
AA	AABB	FC	FABC	HC
⋮	⋮	⋮	⋮	⋮
AaA	AaABb	FAaA	FAaABb	HG
⋮	⋮	⋮	⋮	⋮

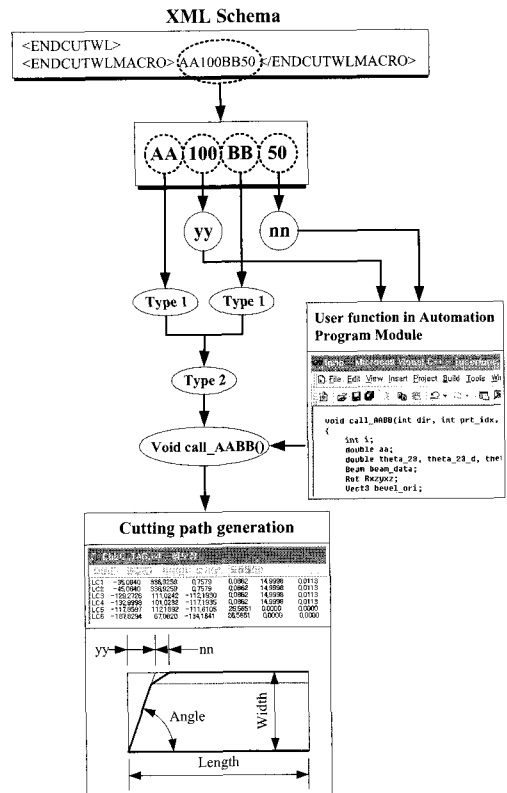


Fig. 13 Example of path generation

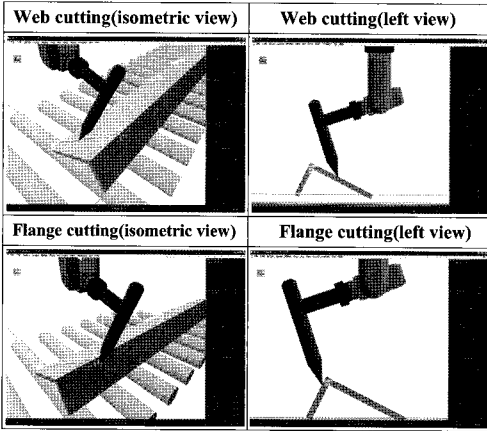


Fig. 14 Example of cutting path generation for 'AABB'

(user function)에서 Type 2에 대한 함수를 불러온다.

6. 'call_AABB' 함수에서 절단 경로의 좌표를 계산한다.
7. 계산된 결과 값을 text 파일 형식으로 지정된 경로에 저장한다.

Fig. 14는 자동화 프로그램 모듈안의 'AABB' 함수를 이용하여 절단 경로를 생성한 후 시뮬레이션 한 결과를 나타낸다.

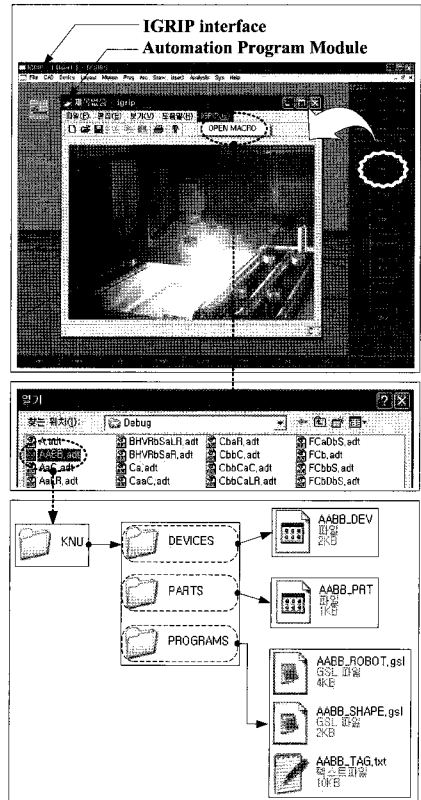
6. 자동화 프로그램 모듈을 이용한 시뮬레이션

자동화 프로그램 모듈의 목적은 macro 데이터에 포함된 각종 수치 데이터를 분석하여 형강 절단 시뮬레이션을 위한 IGRIP용 부품 및 장치 파일을 자동으로 생성하여 시뮬레이션을 자동화하는 것이다. 자동화 프로그램 모듈과 IGRIP이 연동되어 시뮬레이션이 구현되는 과정을 Fig. 15에 나타낸다.

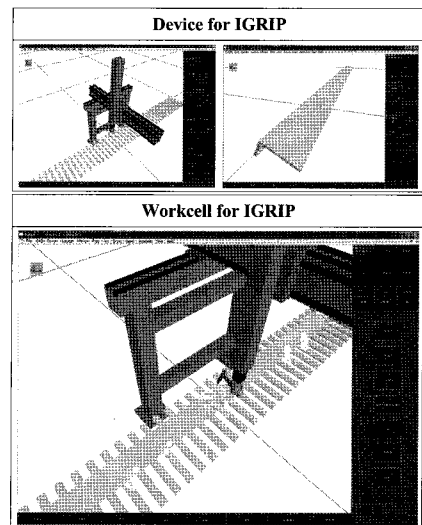
IGRIP의 사용자 버튼인 'XML'을 선택하면 자동화 프로그램 모듈이 실행되고 자동화 프로그램 모듈의 'File menu'에서 'OPEN MACRO'를 선택하여 저장되어 있는 macro 파일을 선택한다. 사용자가 원하는 macro 파일이 선택되면 형강 절단 시뮬레이션에 필요한 부품, 장치, 교시점 및 GSL 프로그램 등이 지정된 경로로 각각의 폴더에 자동으로 생성되어 저장된다. 저장된 각각의 파일들은 형강 절단 시뮬레이션을 위해 사용된다.

7. 결론

본 연구의 목적은 갠트리형 로봇의 형강 절단작업을 위한 시뮬레이션 데이터를 자동으로 생성하는 프로그램 모듈을



(a) Procedure of generating the data using Automation Program Module



(b) Simulation based IGRIP

Fig. 15 Simulation using Automation Program Module

개발하는 것이다. 이를 위하여 첫째, 갠트리형 로봇의 각 부품을 CATIA에서 모델링한 후 IGRIP용 3차원 모델로 변환하였다. 둘째, 자동화 프로그램 모듈을 사용하여 로봇 시뮬레이션 전용 프로그램인 IGRIP의 시각화를 구현하였다. 셋째, 자동화 프로그램 모듈을 이용하여 작업 정보들이 들어있는 macro 파일을 IGRIP 상에서 사용 할 수 있는 부품, 장치 파일로 자동으로 변환되도록 하였다. 넷째, 수치 데이터를 추출하여 text 파일 형식의 교시점이 자동으로 생성되도록 하였다. 마지막으로, 갠트리형 로봇과 형강의 상호 작업을 위해서 ‘갠트리형 로봇 이동 GSL 프로그램’과 ‘형강 이동 GSL 프로그램’이 자동으로 생성되도록 하였다.

후 기

본 연구는 산업자원부에서 지원한 “실시간 3차원 형상 측정 및 고속절단용 로봇 시스템 개발” 사업을 통해 수행되었으며, 이에 관계자 여러분께 감사를 드립니다.

참 고 문 헌

- (1) Kim, S. H., 2007, *High performance hybrid profile cutting robot system*, Final Report of Samsung Heavy Industries CO. LTD, Korea.
- (2) Young, H. C., Han, S. H., and Lee, M. H., 2001, “Development of a 3D Off-Line Graphic Simulator for Industrial Robot,” *KSMTE J.*, Vol. 10, No. 3, pp. 19~25.
- (3) Leslie, B., 1998, “The development of software to assist in off-line programming for robotic fettling of cast components,” *Industrial Robot J.*, Vol. 25, No. 4, pp. 282~287.
- (4) Jeong, D. Y. and Han, S. H., 2002, “Intergrated Control System Design of SCARA Robot Based-On Off-Line Programming,” *KSMTE J.*, Vol. 11, No. 3, pp. 21~27.
- (5) Kim, J. K. and Lee, S. H., 2004, “The Welding/Welding History DB Construction of OLP System For the Multi-Layer Welding on the Thick Steel Plates In Low Speed Diesel Engines,” *KSPE J. Autumn Conference*, pp. 760~763.
- (6) Lee, J. H., Kim, C. S., and Hong, K. S., 2005, “Off-Line Programming in the Shipbuilding Industry : Open Architecture and Semi-Automatic Approach,” *International Journal of Control, Automation, and System*, Vol. 3, No. 1, pp. 32~42.
- (7) Cha, T. I., 2002, “Research on 3D Simulation for Assembly Process of Ship Based on Object-Oriented Information Modeling,” Korea Maritime University, A Thesis for a Master, Republic of Korea.
- (8) Chung, J. H., 2004, “Development of robot welding program for gantry-type robot using neutral CAD data format,” Ulsan University, A Thesis for a Master, Republic of Korea.
- (9) Frank, S. C., 2000, “A Methodology for Developing Robotic Workcell Simulation Models,” *IEEE Trans. J. Proceedings of the Winter Simulation Conference*, Vol. 2, pp. 1265~1271.
- (10) Yang, J. S., Han, S. H., Kim, B. E., and Park, C. C., 2003, “A Macro Parametric Data Representation for CAD Model Exchange using XML,” *KSME J. A*, Vol. 27, No. 12, pp. 2061~2071.
- (11) Mun, S. H., Doh, Y. C., Park, G. B., Kim, D. K., and Kim, S. H., 2006, “A CAD/CAM System for Sub-Assembly Welding Robot System at Shipyards,” *KSME J. Spring Conference*, pp. 440~443.
- (12) Kim, E. J., Rhee, S. Y., Kim, B. S., Park, J. H., and Park, Y. J., 2003, “A Dedicated CAM for Welding Automation System in Assembly Line at Shipyards,” *KSME J. Autumn Conference*, pp. 1757~1762.