

렌즈 조립 셀을 위한 제어소프트웨어 설계

Designing of Control Software for A Lens Assembly Cell

*#강성복¹, 강신가¹, 김형태¹, 강희석¹, 조영준¹, 김진오²

**S. B. Kang(sbkang@kitech.re.kr)¹, S. K. Kang¹, H.T.Kim¹, H.S.Kang¹, Y.J.Cho¹, J.O.Kim²

¹ 한국생산기술연구원 메카트로닉스팀, ² 광운대학교 정보제어학부

Natural Language Script, Command Language Script, Script Analyzer, Control Software Module

1. 서론

조립생산시스템에서 유연하고 민첩한 지능형 생산 시스템의 필요성이 증대 되면서 다품종 대량생산, 저가격화 그리고 초소형 초정밀화가 요구되는 제품을 안정된 품질로 조립생산이 가능한 생산시스템이 요구되어지고 있다. 특히 각 제품의 특성에 따라 동일한 조립 Cell과 Module화 된 장비를 추가하여 공정특성에 적합한 Assembly Cell을 신속하고 빠르게 구축하려는 연구가 국내외에서 활발하게 연구되고 있다. 국내에서는 비록 생산 시스템은 아니지만 RUPI(Robot Unified Platform Initiative)라는 이름으로 로봇의 프로그래밍 환경을 통합하려는 연구가 진행되고 있다. OS나 언어에 종속되지 않고 다양한 형태의 로봇을 하나의 플랫폼으로 제어할 수 있게 하려는 연구이다. 이런 통합 플랫폼은 다품종 대량생산이 요구되는 Lens Assembly Cell에 또한 필요하다.

본 연구는 이런 통합 플랫폼 연구를 Lens Assembly Cell을 대상으로 적용하려는 시도이며, 생산 시스템 중 공정 및 제품의 변화가 심한 Lens Assembly Cell을 위해 로봇 마다 동일한 인터페이스를 가지는 Virtual Machine을 구동하고 공통된 형식의 프로그래밍 방법으로 시퀀스 생성을 가능하게 하였으며 이를 Lens Assembly Cell에 적용하여 시퀀스 생성 시간을 단축시키고자 하였다.

2. 제어소프트웨어를 이용한 시퀀스 생성 방법

2.1 시퀀스 생성 방법 및 구조

하위 레벨의 C++ 코드를 직접 구현하지 않고 제어 소프트웨어를 사용하여 사용자가 일상작업에서 사용하는 작업 중심의 자연어 스크립트를 정의하고 이를 이용하여 시퀀스를 생성한다.

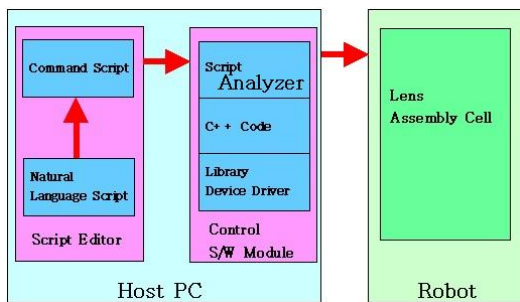


Fig. 1 Script Flow

Fig. 1에서처럼 스크립트 에디터에서 작업 중심의 자연어를 정의하고 이동경로를 입력하면 로봇제어언어에 가까운 명령어로 자동 변환해 주며, 제어소프트웨어 모듈을 통해 로봇을 제어하게 된다.

제어소프트웨어 구조는 크게 세가지로 나눌 수 있다. 자연어 스크립트와 명령어 스크립트를 생성하고 편집하는 스크립트 에디터, 명령어 스크립트를 인식하고 인수를 제어 소프트웨어 모듈에 전달하는 스크립트 분석기, 로봇 제어를 위한 구동 알고리즘이 포함된 제어 소프트웨어 모듈이다.

에디터를 통한 작업의 표현은 Table 1과 같이 일상작업에 사용되는 사용자 중심의 문장구조에서 동사를 강조하는 명령문 형태의 자연어이며 단어 및 구문분석기를 사용하여 이를 로봇제어언

어와 유사한 명령어 스크립트로 변환된다.

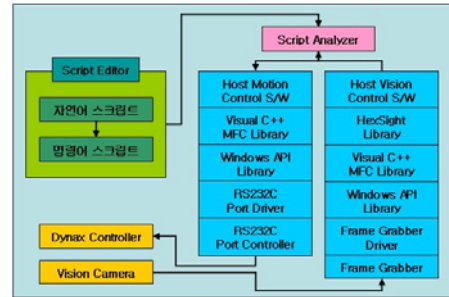


Fig. 2 Structure of Control S/W

스크립트 분석기는 명령어 스크립트의 명령어와 파라미터를 정확하게 구분 인식하여 제어 소프트웨어 모듈에 전달한다.

Table 1 Definition of Natural Language and Command Language

자연어	명령어	설명
Origin Return	ORGR	초기위치로 이동
Approach	MOVEX, MOVEY	목표위치에 접근
	MOVEZ, MOBET	목표위치의 3차원(X,Y,Z) 좌표 입력
Pick-up	GRASP	부품 잡기
Move	MOVEX, MOVEY	목표위치로 이동
	MOVEZ, MOBET	목표위치의 3차원(X,Y,Z) 좌표 입력
Release	UNGRIP	부품 놓기
Identify	INPCT	부품의 판별
Align	ALIGN	부품조립을 위한 정렬
Insert	INSRT	부품을 목표 거리만큼 삽입

2.2 제어 소프트웨어를 이용한 시퀀스 생성 과정

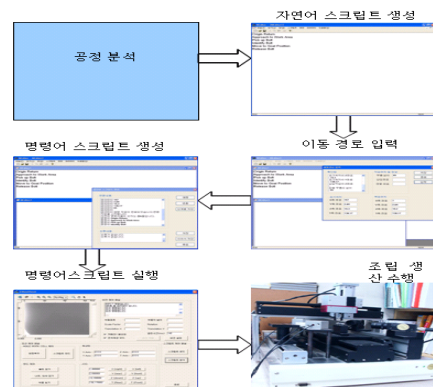


Fig. 3 Programming Process of Control S/W

제어소프트웨어를 이용한 시퀀스 생성과정은 공정분석을 통해 자연어 스크립트를 생성하고 생성된 자연어에는 목표 위치가 포함되어 있지 않으므로 작업을 수행하기 위해 이동해야 하는 위치 정보를 3차원 좌표(X,Y,Z)로 입력한다. 또한 G(Grripper), T(Feed Tray)축을 추가로 설정할 수 있다.

자연어 스크립트와 이동 경로를 저장한 후에 작업 Parameter를 포함한 명령어 스크립트로 변환하게 된다. 이러한 과정을 통해 자연어 스크립트에 축약되어 명시된 작업을 단위작업으로 풀어 명령어를 생성하고 각 명령어는 제어 소프트웨어 모듈에 1:1 대응하여 로봇을 제어하게 된다.

3. 제어소프트웨어를 이용한 Lens Assembly

제어소프트웨어를 Fig. 4와 같이 단축 스테이지 모듈로 이루어진 5축 Lens Assembly Cell에 적용하였다.



Fig. 4 Lens Assembly Cell

제어소프트웨어를 적용할 타겟공정은 Fig. 5의 PCB에 홀더와 렌즈를 결합하는 홀더 Attach, 렌즈 Assembly 공정에 적용하였다.

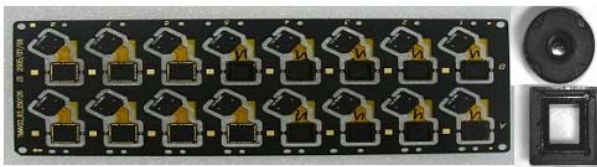


Fig. 5 PCB and Holder, Lens

3.2 스크립트 코드 비교

자연어 스크립트	C++ 코드
Origin Return	(생략)
Approach to Working Area	GripHolder();
Pick-up (Holder/Lens)	InspectPart();
Identify (Holder/Lens)	//검사하고홀더가맞으면다음공정을계속한다.
Move to Goal Tray	if(mVisionData0->mPartState == PS_Holder)
Release (Holder/Lens)	{
명령어 스크립트:	AssembleHolder();
ORGR	};else//잘못된부품이라면반송
MOVEX,167	{
MOVEY,2.69	Return1();
MOVEZ,19.3	break
MOVET,138.17	}
GRASP,65	void GripHolder(void)
INPCT,167,19.3	{
MOVEX,1	//부품트레이로x축이동
MOVEY,2.69	getCHost().setMoveInfo(0, (float)167);
MOVEZ,19.2	getCHost().moveABS_M();
MOVET,138.17	while(getCHost().moveDone() == FALSE)
UNGRP	{
	Sleep(100);
	}
	//홀더를잡기위해z축내림
	getCHost().setMoveInfo(2, (float)19.1);
	getCHost().moveABS_M();
	while(getCHost().moveDone() == FALSE)
	{
	Sleep(100);
	}
	(생략)
	}

Fig. 6 Script Code Comparison

제어소프트웨어는 자연어 스크립트를 명령어 스크립트로 변환하고 명령어 스크립트를 사용하여 제어 소프트웨어 모듈을 동작시킨다. 자연어 스크립트가 일종의 심볼 역할을 하고 명령어 스크립트가 세부 작업을 지시하게 되며 스크립트 분석기와 제어 소프트웨어 모듈을 통해 로봇을 구동한다.

Fig. 6과같이 자연어 스크립트의 예시와 C++ 코드를 비교하면 자연어 스크립트가 더 작은 코드로 로봇을 제어하는 것을 알 수 있다. 또한 숙련되지 않은 사용자도 작업내용을 중심으로 한 자연어를 이용하여 쉽게 로봇을 제어할 수 있고 또한 공정의 변경에 쉽게 대처해 나갈 수 있다.

3.3 스크립트 에디터와 제어 소프트웨어 모듈 GUI

스크립트 에디터와 제어 소프트웨어 모듈 GUI를 살펴보면 Fig. 7, Fig. 8과 같다

에디터는 작업을 생성 및 저장하는 에디터 기능과 이동경로 입력과 명령어 스크립트 변환 기능 등으로 이루어져 있다. 에디

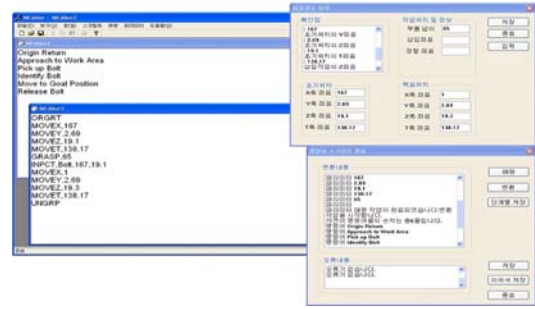


Fig. 7 Script Editor

터 창을 통해 입력한 자연어 스크립트에 이동 경로를 입력한 후 명령어 스크립트로 변환 저장하게 된다. Fig. 9는 제어 소프트웨어 모듈의 GUI로 로봇의 동작시범을 위한 원점 복귀 기능과 각 축을 동작시킬 수 있는 JOG 기능을 포함하고 있다. 또한 비전 제어 패널에서는 카메라를 켜고 끄기 위한 아이콘과 HexSight를 동작시키고 설정을 불러올 수 있는 비전 동작과 비전 설정 버튼을 포함하고 있다. 스크립트 제어를 위해 명령어 스크립트를 불러오고 명령어 스크립트를 정해진 변수에 넣어 로봇을 제어하는 기능을 구현하였다.



Fig. 8 Control S/W Module GUI

4. 결론

제어소프트웨어를 통한 시퀀스 생성방법은 C++ 코드와 비교하여 짧은 코드로 시퀀스를 생성하고 Lens Assembly Cell을 제어할 수 있었다. 그 결과 다품종 대량생산, 저가격화 그리고 초소형 초정밀화가 요구되는 다양한 조립공정에 제어소프트웨어를 적용함으로써 숙련되지 않은 사용자가 쉽게 시퀀스를 생성하고 빠르게 변화하는 모델 및 공정변경에 민첩하게 대응할 수 있을 것으로 예상된다.

참고문헌

- 1 Donald R. Myers and Michael J. Pritchard, Mark D.J. Brown, "Automated programming of an industrial robot through teach-by showing", Proceedings of the 2001 IEEE International Conference on Robotics & Automation Seoul, Korea. May 21-26, 2001
- 2 Wyatt S. Newman, Andy Podgurski, Roger D. Quinn, Frank L. Merat, Michael S. Branicky, Nick A. Barendt, Greg C. Causey, Erin L. Haaser, Yoohwan Kim, Jayendran Swaminathan, Virgilio B. Velasco, Jr., "Design lessons for building agile manufacturing systems", IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 16, NO. 3, JUNE 2000
- 3 Emanuele Carpanzano, Andrea Cataldo, Marco Dona, "Rapid prototyping test-bed of logic control solutions for reconfigurable manufacturing systems", 2005 IEEE
- 4 Chandimal Jayawardena, Keigo Watanabe and Kiyotaka Izumi, "Teaching a Tele-robot using Natural Language Commands", IEEE 2000