

VRML 과 개방형구조를 이용한 사용자 기반의 로봇 시뮬레이션 개발

김창세[†] · 홍금식* · 김수호**

Development of User-Based Robot Simulation using VRML and Open Architecture

Chang-Sei Kim, Keum-Shik Hong and Soo-Ho Kim

Key Words : Robot Simulation(로봇시뮬레이션), VRML(가상현실모델링언어), Open Architecture(개방형구조), Off-line programming(오프라인프로그래밍)

Abstract

Robot simulation technique is essential not only for robot developers to design robotic systems but also for robot operators to predict robot motion, configure system layout, and increase robot ability. However, commercial robot simulation software such as ROBCAD, IGRIP, and so on are expensive and sometimes they are difficult to customize into industrial purpose programming for users. Therefore, user-based simulation programming is required to magnify the efficiency of robot system. In this paper, we show the methodology of developing user-based robot simulation programming using PC(personal computer), Open-Inventor, and Windows Programming. The developed programming has been successfully applied to welding robot systems of a shipbuilding industry. Also, the methodology presented here can be easily extended to simulate manipulators of other typed mechanism on user's PC.

1. 서 론

산업현장에 로봇시스템의 적용이 늘어감에 따라, 더 효과적이고 손쉬운 운영을 위해서, 시스템을 예측 가능한 시뮬레이션 방법과 주변기술에 대한 많은 연구가 진행되고 있다. 조선용 용접로봇의 off-line programming 은 산업현장에서 사용되는 로봇 시뮬레이션의 한가지 예로, 로봇 기구의 시뮬레이션, 작업 계획, 로봇 명령어 생성 및 작업 모니터링까지를 포함한다. 또한, 로봇이나 새로운 구조의 생산시스템 설계자의 입장에서, 기구를 만들기 전에, 시스템의 3 차원 시뮬레이션을 통해 설계와 적용방법을 보완하기 위한 목적으로

다양한 로봇 시뮬레이션 프로그램을 활용한다. 로봇시스템의 경우에는 ROBCAD, IGIP 과 같은 상용 소프트웨어가 실제 많이 사용되고 있으나, 고가이고, 새로운 로봇 언어나 현장설치를 목적으로 적용하는 데는 많은 어려움이 따르며, 특히, 특정 작업대상물에 대한 로봇의 오프라인 프로그래밍 추가와 같이 특정한 기능이 필요로 하는 경우에는 상용 소프트웨어를 적용하는데 많은 제약조건이 따른다. 따라서, 사용자가 요구하는 기능에 부합하는 시뮬레이션을 위해서는, 사용자 기반의 로봇 시뮬레이션 프로그램을 개발하는 것이 필수적이다. 사용자 기반의 시뮬레이션을 위해서는, (1) 요구되는 기능 파악, (2)로봇 기구학 해석, (3)로봇 운동역학 해석, (4)3 차원 시뮬레이션 환경 구성 기술 이 요구된다.

로봇시스템 제어 및 응용을 목적으로 하는 많은 시스템 개발자 및 로보틱스 또는 운동역학을 전공하는 연구자들의 입장에서, 가장 구현하기 어려운 것이 3 차원 시뮬레이션 환경을 구성하는

[†] 부산대학교 기계공학부 제어자동화시스템전공
E-mail : coriolis@naver.com

TEL : (051)510-1481 FAX : (051)514-0685

* 부산대학교 기계공학부 제어자동화시스템전공

** (주)대우조선해양 자동화연구소

것으로, 사용자 기반의 시뮬레이션 개발 시에 가장 많은 시간과 노력을 필요로 하는 작업이다. 여기서는, 이러한 어려움을 해결하기 위해, 사용자 기반의 시뮬레이션 환경 구축 방법을 제안한다.

기술된 방법으로 개발된 시뮬레이션 환경은, 사용자 요구에 부합하는 다양한 기능 구현 목적 및 확장성을 고려하여, VRML (가상 현실 모델링 언어)과 OOP(객체 지향 프로그래밍)를 사용하여 윈도우 환경에서 구성한다. OOP 및 개방형구조의 시뮬레이션 환경은, 몇 개발자들의 동시 작업에 의해 필요한 기능을 라이브러리 형태로 개발/통합할 수 있어, 개발기간을 단축할 수 있다.

논문의 구성은 다음과 같다. 2 장에서는 VRML 및 CAD 데이터로부터 VRML 모델을 얻는 과정을 설명한다. 3 장에서는 2 장에서 얻어진 모델을 기구학식으로 연결하여 로봇 모션을 생성하는 방법을 보인다. 4 장에서는 로봇 명령어 번역 및 기구학식을 이용하여 로봇을 구동하는 방법을 보이고, 5 장에서는 사용자 기반의 시뮬레이션 기법을 조선용 용접 로봇의 오프라인프로그래밍에 적용한 사례를 보인다. 개방형구조로 설계된 시뮬레이션 환경에 조선용 용접로봇의 오프라인프로그래밍에 필요한 기능을 추가하는 방식으로 적용되었다. 6 장에서 결론을 맺는다.

2. VRML 모델 및 CAD 인터페이스

VRML 을 이용한 기계시스템의 시뮬레이션에 대한 연구는 다양한 분야에서 연구되고 있다. 또한, VRML 은 그래픽 디자인 및 인터넷상에서 그래픽 환경을 만들기 위한 방법으로 주변에서 쉽게 접할 수 있다. 3 차원 모델링 도구로, VRML 은 아래와 같은 장점을 가진다.

- (1) 모델링 객체가 표준 문법적인 구조를 가지는 텍스트 구조로 되어있으므로, 해석이 쉽다.
- (2) 다른 CAD 데이터로부터, 모델링 객체 자동 생성/변환 등의 작업이 용이하다.
- (3) 모델링 객체의 기본 단위가 꼭지점이므로, 교시점과 같이 유용한 정보를 쉽게 얻는다.
- (4) VRML 은 인터넷 상에서 표현될 수 있는 모델이므로, 인터넷을 사용한 로봇 모니터링의 목적으로 확장할 수 있다.

특히, 다양한 CAD 데이터로부터 VRML 형상 모델링에 필요한 정보를 선별 추출할 수 있으므로, 복잡한 형상모델을 단순화 하여, 그래픽 부하를

줄일 수도 있다. 또한, 대부분의 3 차원 CAD 프로그램은 설계 디자인을 VRML 로 변환하는 기능을 지원하기 때문에, 시스템 제작을 위해 만들어진 로봇 설계 데이터를 시뮬레이션 모델로 변환하여 사용할 수 있으므로, 시뮬레이션에 필요한 모델링 시간을 줄일 수 있다. 뿐만 아니라, 로봇 또는 기계시스템의 작업대상물도 CAD 로 작업된 데이터를 VRML 로 변환하여 사용할 수 있다. VRML 내부의 꼭지점, 선, 호 등의 정보는 교시점(teaching point)를 생성하는데 활용한다.

CAD 인터페이스는 CAD 데이터로부터 VRML 모델을 추출과 작업대상물의 형상 정보 획득을 목적으로 한다. CAD 인터페이스를 통해 VRML 로 변환된 데이터의 예는 Fig. 1 (a)이고, 작업대상물의 형상정보 중에서 작업에 필요한 교시점 생성을 위해 얻어진 데이터의 예는 Fig. 1 (b)다. Fig. 1 (a)는 로봇이나 기구 시스템의 모델로 활용되고, Fig. 1 (b)는 로봇프로그램을 자동으로 생성하는데 활용된다.

```
#VRML V2.0 utf8
Collision {
  children
  DEF b261_parts0001 Group {
    children
    Shape {
      appearance
      Appearance {
        material
        Material {
          ambientIntensity 0.2
          ...
        }
      }
      geometry
      IndexedFaceSet {
        coord
        Coordinate {
          point [ 60.39 11.52 0.5,
                60.39 11.53 0.5,
                ...
          ]
          normalPerVertex TRUE
          coordIndex [ 8, 7, 6, 5, 4, -1,
                    2, 261, 260, 1, -1,
                    3, 262, 261, 2, -1,
                    ...
          ]
        }
        rotation 0 0 1 0
      } translation -60.39 -11.5231 -0.496971
    }
  }
}
```

(a)VRML model obtained by CAD interface

```
BEGIN_INFO
'S6G9';
BEGIN_JOINT
J-1; 527-FR86A-1;527-FR86A-S1;15.5;18.0;
BEGIN_WELD
W-1;H:0.707;-0.036;-0.706;525;
BEGIN_SEGMENT
1836,-25,314; 0,0,0 1836,500,314;
END_SEGMENT
END_WELD
END_JOINT
END_INFO
```

(b)Workpiece data to generate robot programs

Fig. 1 An example of CAD interface

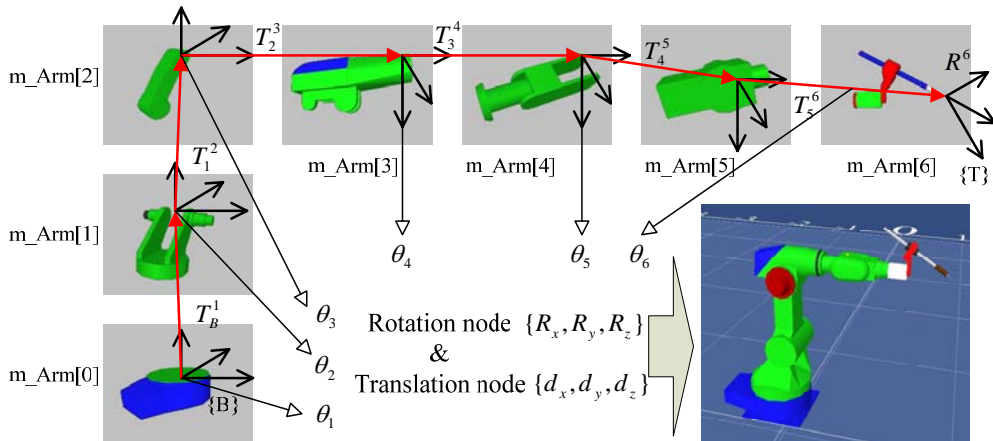


Fig. 2 Construction example of 6-axis robot body on 3D graphic environment using kinematics

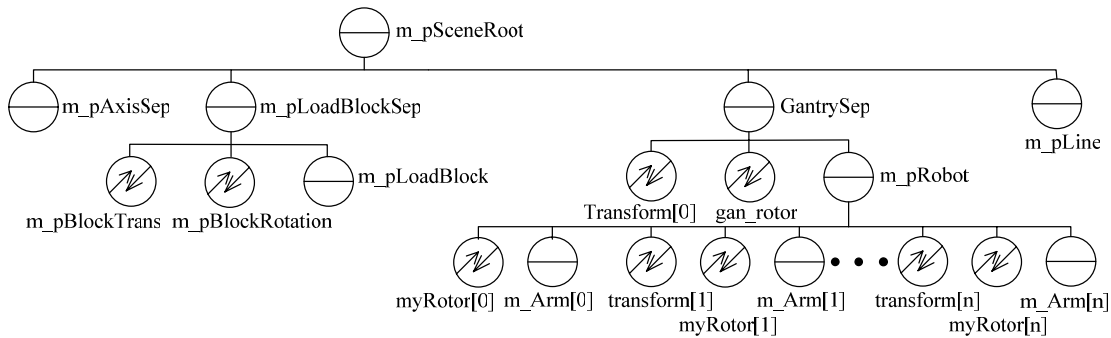


Fig. 3 Construction of simulation node including model node, motion node, and auxiliary graphic node

3. 로봇 시뮬레이션 구성

일반적인 6 축 로봇의 기구학 식을 이용하여, 3 차원 로봇 모델을 동작시키기 위해서는, 로봇 기구부 모델, 모델을 이동시키는 병진/회전 노드, 및 기구학/동역학 계산부분의 3 가지가 필요하다. 로봇 기구부 모델은 앞서 설명한 VRML 변환으로 얻을 수 있다. 병진/회전 노드는 각 모델의 상위에 위치하여 기구/동역학 계산식으로부터 얻어진 샘플링 시간단위의 병진이동거리 또는 회전각 값을 입력 받아, 병진/회전 노드의 하위에 위치한 모델을 이동시키는 역할을 한다. 기구학/동역학 식은 일정 시간단위로 실행되는 스래드(thread)를 통해 한 스텝씩 계산되는데, 실제 제어기의 제어 샘플링과 스래드의 인터럽트 시간을 일치시킨다면, 실제 로봇의 모션과 동일한 시간단위로 실시간 시뮬레이션이 가능하다.

Fig. 2 와 Fig. 3 은 각각 6 축 로봇의 기구학 식을 이용하여 로봇 바디를 구성하는 방법과 로봇을 구동하는 노드(node)를 설계하는 방법을 보여준다. Thread 내부에서 계산된 기구학식 또는

동역학식의 계산 결과인 회전각 또는 변위는 Fig. 3 과 같이, 각 모델 노드(m_Arm[0], ..., m_Arm[n]) 의 상단에 위치한 transform node 의 변수로 입력되고, 이는 모델을 결과치 만큼 이동시키는 방식으로 로봇의 동작이 구현된다. Transform node 는 translation 과 rotation 으로 구성된다. 매 인터럽트 시간마다 모션이 업데이트 되므로, 컴퓨터 상에서는 실시간으로 로봇이 동작한다.

4. 로봇 명령어 번역 및 시뮬레이션

개발자가 정해놓은 명령어 또는 로봇 제작 업체에서 제공하는 명령어를 로봇 언어로 정의한다. 로봇 언어는 통상 사용자가 이해하기 쉬운 레벨의 명령어와 각 명령어의 실행에 필요한 파라미터로 구성된다. 그러나, 실제 로봇 구동 시에는, 각 명령어가 지시하는 모션을 생성하기 위해, 제어기에서는 명령어를 파싱(parsing)하고, 각 명령어에 해당되는 제어기 내부 프로그램의 함수와 입력된 기준위치정보등을 이용하여 로봇을 제어하는 기능이 수행된다. 시뮬레이션 환경 구축시에도, 제어기의 동작과 마찬가지로, 방법으로,

각 로봇 명령어를 파싱하고 명령어에 해당하는 동작을 생성하고 제어하는 루틴을 구현해야 한다. 상용로봇의 경우에는 실제 로봇의 가능 동작을 바탕으로 동작을 프로그래밍 해야 하는 불편함이 있지만, 새롭게 개발되는 로봇 및 제어기의 경우에는 개발되는 로봇 제어기의 프로그램을 이용하여 시뮬레이션이 가능하다. 즉, 로봇 시스템 개발자라면, 제어기 내부의 로봇 제어프로그램을 시뮬레이션 프로그램에 가져와서, 실제 제어기와 동일한 기능을 구현할 수 있다. 또한, 로봇 운동역학이 반영되어 있다면, 작업대상물과의 충돌, 작업시간, 모션제어성능 등을 실제 시스템 구성 전에 3차원 시뮬레이션을 통해 확인할 수 있다.

하나의 로봇 작업이 로봇의 여러 가지 동작으로 이루어지는 경우, 각 동작에 해당하는 몇 줄의 명령어가 한 개의 파일을 구성하게 되는데, 이를 로봇프로그램이라 한다. 다양한 형상과 크기를 갖는 작업대상물에 능동적인 로봇 시스템 적용을 위해, 표준프로그램과 작업프로그램으로 구분하는 방법을 사용한다. 표준프로그램은 작업대상물의 형상 및 크기 정보가 반영되지 않은, 실제 작업 이전 단계의 로봇프로그램이고, 작업프로그램은 작업대상물 또는 TCP(tool center position) 이동 거리 등의 실제 수치가 반영된 로봇프로그램이다. 조선용 용접로봇과 같이, 다양한 형태, 크기의 작업대상물을 가지는 로봇 시스템의 경우, 1 개의 표준 프로그램에 대해, 크기에 따라, 약 50 개 이상의 작업프로그램이 만들어진다. 통상, 로봇 프로그램은 로봇 교시기(teaching pendant)를 이용하여 온라인교시를 통해 만들어왔다. 그러나, 컴퓨팅 및 오프라인프로그래밍 기술이 비약적으로 발전함에 따라, 작업대상물의 형상정보만으로, 자동교시 및 로봇프로그램생성이 가능하게 되어, 작업자에 의존하지 않는 로봇 작업이 가능하다. 작업프로그램 자동생성 알고리즘은 Fig. 4 와 같다.

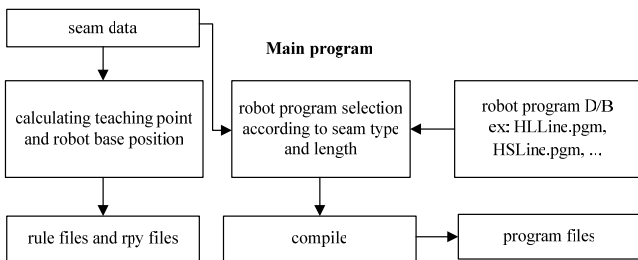


Fig. 4 An algorithm for automatic generation of robot programs

5. 오프라인 프로그래밍

오프라인 프로그래밍 (off-line programming)은 온라인교시(on-line teaching)의 반대 개념으로, 시뮬레이션 또는 프로그래밍 기술을 통해 실제 로봇을 구동하지 않고, 로봇 작업프로그램을 만드는 작업을 의미한다. 대표적인 장점으로,

(1) 다양한 디버깅 도구를 이용하여 효과적인 프로그램이 가능하다.

(2) 시뮬레이션을 통해, 실제 로봇을 구동하지 않고도 모션의 검증이 가능하다.

(3) 표준프로그램 및 작업프로그램의 개념을 이용하여, 한번 생성된 프로그램을 다양한 사이즈, 형상에 대해 쉽게 재사용 가능하다.

(4) 실제 생산과 별개로, 사무실에서도 작업이 가능하므로, 프로그래밍에 의한 생산차질이 없는 것이다.

이러한 장점으로 인해, 오프라인프로그래밍에 대한 많은 연구가 진행되어 왔으며, 새로운 로봇이 등장하거나, 로봇 적용 대상이 늘어감에 따라, 각 현장에 적합한 오프라인 프로그래밍을 만들어주는 연구는 계속 되고 있다.

오프라인 프로그래밍의 효과적인 적용 사례는 다양한 형상과 크기의 로봇 작업대상물을 가지는 조선용 용접로봇에서 찾을 수 있다. 조선 산업에서, 자동화 및 로봇 적용은 조선공정에서 중요한 위치를 차지한다. 특히, 조선산업의 활황에 따른 생산성 향상, 근로자 작업환경 개선에 따른 근골격계 문제, 안전 문제 등을 해결하기 위한 방법으로 그 중요성은 더욱 커지고 있다. 조선산업에 적용된 로봇 시스템의 대표적인 예는 용접로봇이다.

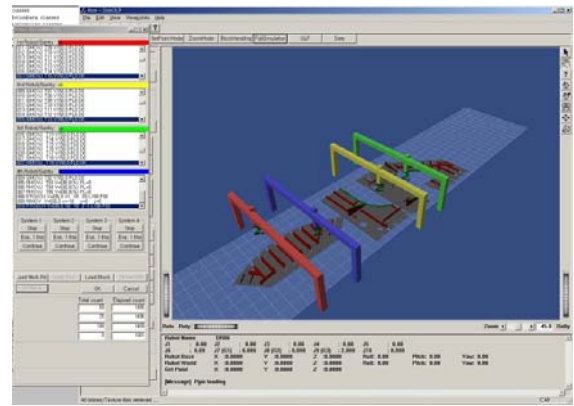
여기서는, VRML 을 이용한 시뮬레이션 기법을 대조립 및 소조립 용접용 로봇의 오프라인 프로그래밍에 적용한 예를 보인다. 대조립 용접용 로봇은 로봇 바디가 바닥에 놓인 상태에서 전면부 용접을 하는 목적으로 적용되고, 소조립 용접용 로봇은 6 자유도의 로봇 바디가 3 축 갠트리 크레인에 매달린 형태로 자동 이동하는 방식으로 부재 사이를 지나다니며 작업한다.

로봇 작업 효율을 높이고 사용자기반의 시뮬레이션환경 개발을 위해 기존의 오프라인 프로그래밍에 추가된 기능으로는, (1) CAD 데이터로부터 얻어진 블록 정보를 가공하여 교시점 및 작업 프로그램을 자동으로 생성하는 기능,

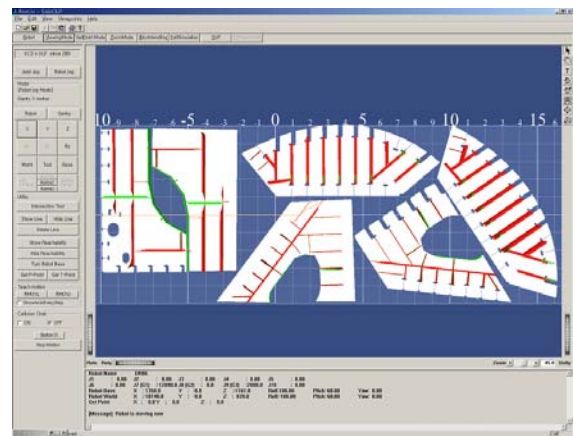
(2)블록배치 시뮬레이션 (3)협업 생산 시뮬레이션, (4)작업시간을 단축하기 위해 유전자 알고리즘을 사용한 최적경로생성, (5)로봇프로그램 편집기능, (6)로봇바디/토치 변경 시뮬레이션, (7)충돌체크 등이 있다.

특히, 소조립 로봇은 갠트리 크레인에 의해 자동으로 다음 작업 위치를 찾아가는 방식으로, 여러 대가 한 정반에서 동시에 작업하게 되므로, 협업(cooperating)시의 로봇 충돌 등에 대한 시뮬레이션은 필수적이다.

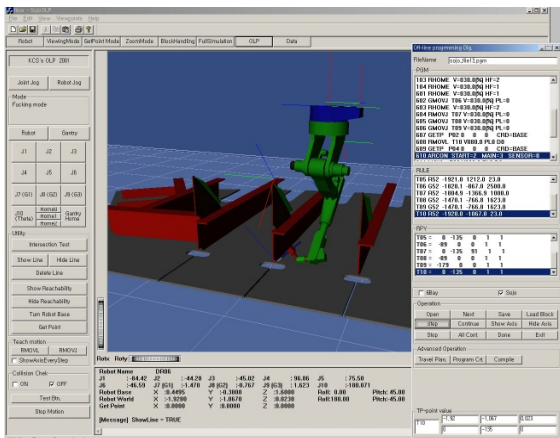
개발된 오프라인 프로그래밍을 적용하여, 로봇 작업준비시간을 기존의 온라인 교시 및 오프라인 프로그래밍 사용 시에 비해 현저히 줄였으며, 이는 작업능률의 향상으로 이어졌다. 또한, 개발된 오프라인 프로그래밍은 유사한 기구학식을 가지는 다양한 크기의 로봇에 대한 시뮬레이션이 가능해져서, 로봇 선정, 로봇 작업 가능영역 판단, 새로운 로봇의 효율성 예측 등의 목적으로도 유용하게 활용하고 있다.



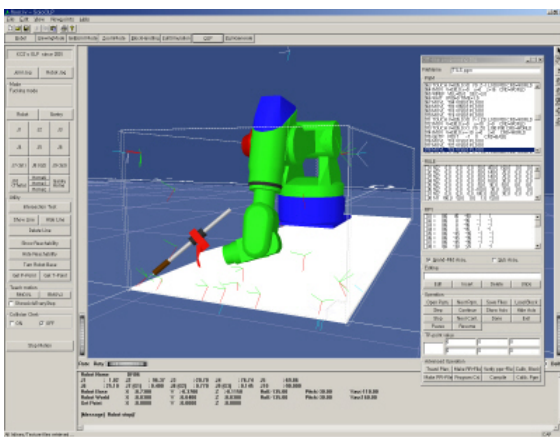
(c) Cooperating welding robot simulation



(d) Block arrangement simulation



(a) Welding robot simulation for component block



(b) Welding robot simulation for grand-assembly

Fig. 5 User-based simulation programming for off-line programming of shipbuilding industry

6. 결론

다양한 형태의 새로운 로봇 시스템이 개발됨에 따라, 이에 대응할 수 있는 시뮬레이션 환경이 필요하다. 특히, 사용자의 요구와 목적을 만족시키는 시뮬레이션을 위해서는 사용자 기반의 시뮬레이션 환경 개발이 필요하다. 이 논문에서는, 사용자 기반의 로봇 및 자동화 시스템 모델링 기법을 제시하고, 이를 현장에서 효과적으로 이용한 사례로 조선용 용접로봇의 오프라인 프로그래밍의 개발을 보였다. 제안된 사용자 기반의 시뮬레이션 환경은 작업의 편리성을 높일 뿐 아니라, 개방형구조로 인한 확장성의 편리함으로, 시스템 배치 등 로봇 및 자동화 연구 분야에 있어서, 다양한 목적으로 응용 개발 될 수 있다.

후 기

이 논문은 2007 년 교육인적자원부의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (지방연구중심대학육성사업/차세대물류 IT 기술연구사업단)

참고문헌

- (1) Aiyama, Y. and Tanzawa, H., 2003, "Management system for distributed and hierarchical robot groups," *Advanced Robotics*, Vol. 17, No. 1, pp. 3-20.
- (2) Borm, J. H. and Choi, J. C., 1992, "Improvement of local position accuracy of robots for off-line programming," *KSME International Journal*, Vol. 6, No. 1, pp. 31-38.
- (3) Buchal, R. O., Cherchas, D. B., Sasami, F. and Duncan, J. P., 1989, "Simulated off-line programming of welding robots," *International Journal of Robotics Research*, Vol. 9, No. 3, pp. 31-43.
- (4) Carvalho, G. C., Siqueira, M. L. and Absi-Alfaro, S. C., 1998, "Off-line programming of flexible welding manufacturing cells," *Journal of Materials Processing Technology*, Vol. 78, No. 1-3, pp. 24-28.
- (5) Choi, M. H. and Lee, W. W., 2003, "Quantitative Evaluation of an Intuitive Teaching Method for Industrial Robot Using a Force / Moment Direction Sensor," *International Journal of Control, Automation, and Systems*, Vol. 1, No. 3 pp.395-400.
- (6) Kim, C. S., Hong, K. S. and Han, Y. S., 2005, "PC-Based Off-Line Programming in Shipbuilding Industry: Open Architecture", *Advanced Robotics*, Vol. 19, No. 4, pp. 435~458.
- (7) Craig, J. J., 1986, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley.
- (8) Ferretti, G., Filippi, S., Maffezzoni, C., Magnani, G. and Rocco, P., 1999, "Modular dynamic virtual-reality modeling of robotic systems," *IEEE Robotics and Automation Magazine*, Vol. 6, No. 4, pp. 13-23.
- (9) Kobayashi, N., Ogawa, S. and Koibe, N., 2001, "Off-line teaching system of a robot cell for steel pipe processing," *Advanced Robotics*, Vol. 12, No. 3, pp. 327-332.
- (10) Nagao, Y.; Urabe, H.; Honda, F. & Kawabata, J. , 2000, "Development of a panel welding robot system for subassembly in shipbuilding utilizing a two-dimensional CAD system," *Advanced Robotics*, Vol. 14, No. 5, pp. 333-336.
- (11) Ogasawara, T., Hashimoto, K., Tabata, M., Komatsu, M., Hara, T. and Kanjo, Y., 1998, "Application of multi-robots control technology to shipbuilding panels," *Journal of the Robotics Society of Japan*, Vol. 16, No. 1, pp. 46-47.
- (12) Richard, G., 1997, "Object-oriented programming for robotic manipulator simulation," *IEEE Robotics and Automation Magazine*, Vol. 4, No. 3, pp. 21-29.
- (13) TECNOMATIX Technologies, 1998, *ROBCAD User Manual*, TECNOMATIX.
- (14) Wernecke, J., 1994, *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor*, Open Inventor Architecture Group.