

Pyro를 사용한 지능형로봇 시스템 시뮬레이션 환경구축 사례연구

(A Case Study on Building AI robotics Simulation environment using Pyro)

이병준⁰, 우균

부산대학교 컴퓨터공학과

imjuni@pusan.ac.kr, woogyun@pusan.ac.kr

ByungJoon Lee⁰, Gyun Woo

Dept. of Computer Engineering, Pusan University

요 약

지능형로봇 산업은 큰 성장 잠재력을 가진 신산업이다. 하지만 로봇을 개발하기 위해서 설계, 구현 단계에서 실제 로봇을 사용하기 어려워 시뮬레이션 환경이 많이 사용된다. Pyro 프로젝트는 파이썬 언어로 작성된 로봇 시뮬레이션 및 제어 환경이다. Player/Stage 프로젝트는 C와 C++로 작성된 로봇 시뮬레이션 및 제어 환경이다. Pyro 프로젝트는 Player/Stage 통합환경을 제공하여 Pyro를 사용하는 것과 같이 Player/Stage를 사용할 수 있다. 본 논문에서는 Pyro와 Stage 시뮬레이터를 이용한 시뮬레이션 환경을 구축하고 적용사례 등을 서술한다.

1. 서 론

지능형로봇 산업은 큰 성장 잠재력을 가지고 있으며 기술혁신과 신규 투자가 유망한 신산업이다. 산업자원부에서 발표한 “지능형로봇산업 비전 및 발전전략”에 따르면 로봇산업은 2020년경 국내시장규모 100조원을 달성할 것으로 예측되며 첨단 신기술 분야의 복합체로 신산업의 창출을 촉진하는 Robot Convergence¹ 연구가 활발히 이루어지고 있다[1]. 또한 출산을 감소, 고령화 사회 진입에 따라 노동력을 대체할 수 있는 미래산업으로 전망되고 있다. 현재 국내 로봇산업은 3,500억 원으로 세계 6위 수준이며 반도체산업과 공작기계 부분의 로봇을 포함할 경우 1조 4,000억 원 규모로 추정된다.

로봇을 설계, 개발하기 위해서 실제 로봇을 사용하여 개발하면 개발비용이 높고 따라서 실패에 따른 위험부담이 크다. 이를 보완하기 위해서 범용의 목적을 지닌 확장 가능한 다양한 로봇개발 플랫폼과 로봇 시뮬레이션 환경이 각광받고 있다.

Pyro[2]는 Python Robotics의 약자로 인터프리터 언어인 파이썬을 사용하여 개발되었다. 파이썬은 이미 교육과 개발의 용도로 많이 사용되고 있으며 직관적이며

가독성이 좋은 소스코드를 생성해 낼 수 있다[3]. 또한 파이썬은 다른 언어와 합성하여 사용하는 합성프로그래밍 환경이 우수하여 지금까지 개발된 다양한 로봇제어 API를 통합하기 쉽다. 사용자가 쉽게 배워 로봇과 AI 에이전트에 사용될 제어 프로그램을 작성할 수 있는 환경 구축이 Pyro 프로젝트 목표이다. 이를 위해 윈도우 환경이 HAL을 사용하여 장치 드라이버를 추상화 한 것과 유사하게 저수준 로봇 제어 코드를 추상화하였다. 따라서 사용자가 로봇개발에 있어서 저수준 로봇제어 지식이 없어도 사용할 수 있고, 다양한 종류의 로봇에 대하여 소스를 변경하지 않고 사용할 수 있다[4].

Player/Stage 프로젝트[5]는 각각 1999년과 1998년부터 지속적으로 개발된 로봇 및 센서 제어 소프트웨어이다. Player 프로젝트는 TCP/IP와 802.11 무선 이더넷을 기반으로 분산 로봇 제어를 가능하게 하는 로봇 제어 소프트웨어 개발 프로젝트이다. Stage 프로젝트는 Player를 사용하여 이동 로봇 개발자가 자주 사용하는 로봇과 센서, 주변환경들에 대하여 시뮬레이션을 할 수 있도록 개발된 시뮬레이터 소프트웨어이다. Stage 시뮬레이터를 이용한 시뮬레이션을 통하여 로봇 개발 비용을 절감할 수 있고 작성한 로봇과 로봇제어 프로그램의 완성도를 높일 수 있다[6]. 그리고 이를 바탕으로 로봇을 개발하기 위한 개발기간과 비용을 단축할 수 있다.

¹ 지능로봇기술은 메카트로닉스기술을 기반으로, IT, BT 및 NT 관련 요소기술들이 융합되어 구현되는 기술로서 로봇산업 외에 정보통신, 바이오, 나노산업 등으로 효과 파급

본 논문에서는 Pyro 프로젝트와 Player/Stage 프로젝트를 사용하여 로봇 시뮬레이션 환경을 구축하고 예제 프로그램을 구동하여 본다.

2. 시뮬레이션 환경 구축

2.1 Player/Stage 이미지 구축

Player/Stage 프로젝트는 유닉스 또는 리눅스 환경에서 구동된다. 본 논문에서는 데비안리눅스 사용하여 실험을 진행하였다.

VMWare 환경은 윈도우 환경을 사용하는 사용자가 유닉스 또는 리눅스 환경을 운영체제로 설치하지 않고 가상머신으로 유닉스 또는 리눅스 환경을 설치할 수 있도록 해주는 프로그램이다. 현재 VMWare Player와 VMWare Server 버전이 공개되어 있다. 본 논문에서는 VMWare Server 버전을 사용하여 데비안 리눅스 환경 이미지를 생성하였다.

Player/Stage 프로젝트는 독립적으로 구동되는 로봇 제어 및 시뮬레이션 환경이나 여러 GNU 공개 라이브러리를 이용하기 때문에 사전 설치해야 하는 라이브러리가 존재한다. 이러한 라이브러리를 설치하지 않는다면 Player/Stage에서 가지는 의존성 관계에 의해서 설치가 실패하거나 일부 기능을 사용할 수 없다. 필수적인 설치 라이브러리는 표1과 같다.

[표 1] Player/Stage 사전 설치 라이브러리

이름	설명
pkg-config	Automake 및 Autoconf를 위한 라이브러리
GDAL	지리정보 추상화 라이브러리
GSL	과학연산 보조 라이브러리
OpenCV	공개 컴퓨터 비전 라이브러리

사전 설치 라이브러리를 설치 후 Player와 Stage를 설치한다. 데비안의 경우 Precompiled-package 라이브러리가 있으나 의존성 문제로 설치가 되지 않을 수 있다. 버전이 낮은 사전 설치 라이브러리를 사용할 때 주로 발생한다. 이런 경우에는 소스를 다운받아 컴파일 하면 된다. 직접 컴파일 하는 경우 configure 단계에서 특별히 주어지는 옵션은 없지만 의존성 관계를 확인하는 단계에서 실패하는 내용이 없는지 잘 확인해야 한다. 먼저 Player를 설치하고 나서 Stage를 설치하면 된다. 설치를 완료하면 독립적으로 Stage를 구동할 수 있다[4]. 본 논문에서는 Stage를 독립 구동하지 않고 Pyro 플랫폼과 연동하여 사용한다.

2.2 Pyro 및 관련 라이브러리

Pyro는 파이썬 모듈을 설치하는 것과 같이 진행된다. SWIG를 사용하여 파이썬으로 작성하지 않은 부분을 컴파일 하는 단계를 지나고 나면 사용자가 직접 파이썬 모듈이 설치되는 디렉토리로 복사 또는 링크를 생성해야 한다. 현재 공개된 Pyro 최신버전은 4.9.5이다.

Pyro 4.9.5 버전은 Player/Stage 프로젝트 최신버전과 호환이 되지 않는다. 이는 Player/Stage 프로젝트 개발 주기가 Pyro보다 빨라서 업데이트되지 않았기 때문이다[5]. 그래서 Player/Stage 프로젝트 최신버전을 사용하기 위해서는 CVS를 사용하여 Pyro 최신버전을 다운로드 하여 설치하면 된다. 본 논문에서는 CVS를 사용하여 Pyro 5.0.0 버전을 다운로드 하여 사용하였다.

Pyro 프로젝트도 Player/Stage 프로젝트와 마찬가지로 사전 설치해야 하는 라이브러리들이 존재한다. 예를 들면 카메라 기능을 시뮬레이션 하기 위해서는 Stage에서 전달되는 로봇 전방에 위치한 객체 정보를 분석하여 이미지로 표현하는데 이미지로 표현하기 위해서는 이미지 생성관련 라이브러리가 필요하다. 또한 비전 처리를 위해서 파이썬 OpenCV도 필요하다. Player/Stage 프로젝트와 유사하게 일부 라이브러리는 설치하지 않아도 설치가 되지만 기능이 제한된다. 예를 들면 이미지 관련 라이브러리를 설치하지 않으면 카메라 시뮬레이션이 불가능하며, 파이썬 OpenCV를 설치하지 않으면 비전기능을 이용할 수 없다. 보다 완벽한 시뮬레이션을 위해서 필요한 라이브러리는 표2와 같다.

[표 2] Pyro 사전 설치 라이브러리

이름	설명
GNOME XML Library	XML 처리 라이브러리
JPEG runtime Library	JPEG 처리 라이브러리
파이썬 OpenCV	파이썬용 OpenCV
Extended image format support for TCL/TK	Tkinter용 이미지 확장 라이브러리
파이썬 Imaging Library ImageTK	파이썬 Tkinter용 이미지 처리 라이브러리

Pyro를 컴파일 하기 전에 먼저 configure.py파일을 실행하여 Pyro를 컴파일 하기 위한 옵션 설정을 한다. 본 논문에서는 Stage와 Pyro를 병행사용하기 위해서 표3과 같이 Stage 비전 지원 부분과, 카메라 장비를 시뮬레이션 하기 위해서 카메라 관련 부분을 설정하였다.

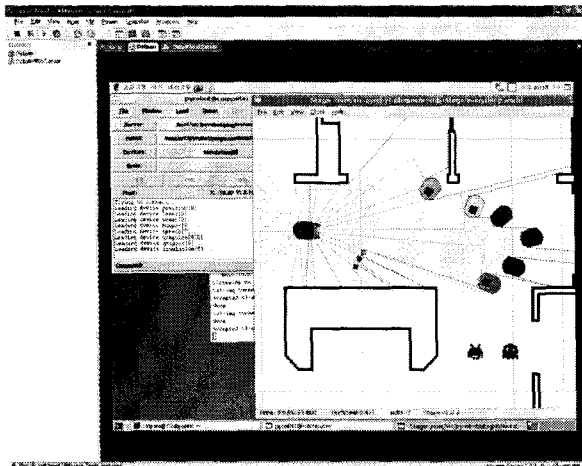
[표 3] Pyro 주요 옵션

Image Processing	영상처리 지원여부
Video for Linux	동영상기록 지원여부
Stage Simulated Vision	비전을 이용한 Stage 시뮬레이션 지원여부
Aibo Vision	아이보 비전 라이브러리 사용여부
Gazebo Simulated Vision	비전을 이용한 Gazebo 시뮬레이션 지원여부
Self-Organizing Map	로봇을 이용한 지도 제작 시뮬레이션 기능 지원여부
Cluster Analysis Tool	클러스터 분석도구 지원여부

단, 마지막에 질문하는 Cluster 분석 도구는 사용하지 않음으로 설정해야 한다. Pyro 5.0.0 버전은 Cluster 분석 도구가 아직 개발이 완료되지 않아 컴파일시 실패하게 된다. 이렇게 옵션에 관한 부분을 설정하고 나면 make를 실행하면 C 또는 C++로 작성된 코드를 컴파일하고 SWIG를 이용하여 파이썬 코드에 연결한다. 이후 파이썬 확장모듈이 있는 위치로 디렉토리를 링크를 만들거나 이동하여 설치를 마무리한다.

Pyro는 복잡한 설치과정을 생략하고 부팅과 동시에 Pyro와 Player/Stage를 사용할 수 있는 LiveCD를 제공하고 있다. Knoppix 기반으로 제작된 이 LiveCD는 설치과정 없이 간편하게 사용할 수 있지만 한글환경이 지원되지 않는 단점이 있다.

그림1은 VMWare Server를 사용하여 데비안 이미지를 생성하여 Player/Stage 및 Pyro를 설치하여 Pyro와 Stage를 구동한 모습이다.



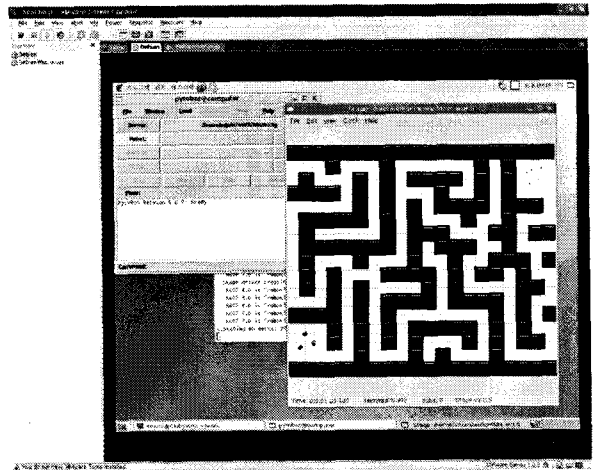
[그림 1] Pyro와 Stage 구동화면

3. 사례연구

Pyro는 로봇 제어 프로그램을 쉽게 제작할 수 있는 환경을 제공하는 것이 목표로 개발되고 있다. 이러한 목표는 로봇에 대한 지식을 배제하고 로봇 인공지능 분야를 학습하기 좋은 도구가 된다[7]. 이미 많은 대학에서는 이런 Pyro가 가지는 특징을 파악하고 로봇 인공지능 분야를 교육하기 위한 효율적인 도구로 활용하고 있다. Pyro 공식 홈페이지를 방문하면 Pyro를 사용하는 50여 곳의 대학 및 단체를 볼 수 있다. 또한 공식 홈페이지는 Pyro를 13가지 모듈로 구분하고, 이를 활용한 예제 커리큘럼을 제공하고 있다.

고등학생을 대상으로 진행하는 연구에 Pyro를 사용하여 진행한 결과, 직관적이고 이해하기 쉬운 문법을 가진 파이썬과 쉬운 사용법을 가진 Pyro는 학생들에게 이해하기 쉽고 흥미로운 도구가 되었다.

그림2는 학생 들이 Pyro를 사용하여 실험한 모습이다. 학생들은 직접 지도를 생성하여 미로를 만들고 목적지에 도달하면 최단 경로로 목표물을 협동하여 나르는 로봇 제어 프로그램을 작성하였다. Pyro는 다양한 방식으로 로봇 제어 프로그램을 작성할 수 있으며 이로 인한 교육적 효과가 높았다



[그림 2] 군집 로봇 시뮬레이션

4. 결론

지금까지 Pyro와 Player/Stage를 사용하여 지능형 로봇 시뮬레이션 환경을 구축한 사례에 대하여 살펴보았다. 본 논문에서 구축하여 사용한 시뮬레이션 환경의 장점은 다음과 같다.

- Pyro 및 Player/Stage는 모두 오픈 소스 프로젝트로 별도의 프로그램 구매를 하지 않더라도 사용할 수 있다. 이는 개발 비용을 절감하는 효과가 있다.

- 신뢰성 있는 환경이다. Pyro 및 Player/Stage는 다양한 주제의 논문이 많이 발표되었으며 공식 홈페이지에서 찾아 볼 수 있다.
- 여러 곳에서 Pyro 및 Player/Stage를 사용하고 있어 정보를 얻기가 쉽다.

본 논문에서 구축된 지능형 로봇 시뮬레이션 환경의 개선방향은 다음과 같이 생각해 볼 수 있다. 현재 Player/Stage가 가지는 강력한 기능 들을 Pyro에서는 완벽하게 사용할 수는 없다. Player가 지원하는 다양한 로봇 플랫폼을 Pyro는 아직 완벽하게 지원하지 못하고 있으며, 개발주기 또한 Player/Stage에 비해서 조금 느리다.

향후 Pyro를 직접 개선하여 Player가 지원하는 다양한 로봇 디바이스를 지원하도록 개선하고 이를 통해 실제 로봇에 적용한다면 이동로봇 개발에 있어서 적은 비용으로 최대의 효과를 내는 로봇 개발 플랫폼으로 사용할 수 있을 것이다.

참고문헌

- [1] 산업자원부: 지능형로봇산업 비전 및 발전전략, 2005년 12월
- [2] Python Robotics Pyro: <http://pyrorobotics.org/>
- [3] Jeffrey Elkner, "Using Python in a High School Computer Science Program", 8th International Python Conference
- [4] Douglas Blank, Deepak Kumar, Lisa Meeden, Holly Yanco, "The Pyro toolkit for AI and robotics", To appear in AI Magazine
- [5] Player/Stage Project: <http://www.playerstage.org>
- [6] Brian Gerkey, Richard T. Vaughan and Andrew Howard. "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems". In Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), pages 317-323, Coimbra, Portugal, June 2003
- [7] Blank, D.S., Kumar, D., Meeden L., and Yanco, H. (2004) Pyro: A Python-based Versatile Programming Environment for Teaching Robotics. *Journal of Educational Resources in Computing (JERIC)*