제33회 한국정보처리학회 춘계학술발표대회 논문집 제17권 제1호 (2010. 4)

효과적인 로봇 프로그래밍 교육을 위한 Pyro 플랫폼 비교

송주원*, 우균*
*부산대학교 컴퓨터공학과
e-mail:{jwsong, woogyun}@pusan.ac.kr

Pyro platform comparison for effective education of robot programming

Ju-Won Song*, Gyun Woo*
*Dept. of Computer Engineering, Pusan National University

요 약

인간이 하기 힘들거나 번거로운 작업은 지능형 로봇이 대체하고 있다. 하지만 로봇을 개발하기 위해서 설계, 구현 단계에서 실제 로봇을 사용하기 어려워 시뮬레이션 환경이 많이 사용된다. 실제 로봇을 사용할 경우 시간과 비용이 많이 들며 개발에 실패하거나 문제가 생겼을 경우 위험부담이 크다. 그러므로 위험부담을 줄이고 개발기간을 단축하기 위해서 실제 구성될 환경과 동작 환경을 고려한 시뮬레이션 환경이 로봇 제어 프로그램 개발에 많이 사용되고 있다. Pyro는 교육과 개발의 용도로 많이 사용되고 있으며, 로봇에 대한 세부적인 지식이 없더라도 제어 프로그램을 충분히 구현할 수 있어 시뮬레이션 환경으로 적합한 로봇 개발 플랫폼이다. 본 논문에서는 Pyro에 대해서 알아보고 Pyro 플랫폼들을 비교해본다.

1. 서론

20세기, 컴퓨터가 발명되면서 사람들의 삶은 혁신적으로 바뀌게 되었다. 인간이 하기 힘들거나 번거로운 많은 작업을 컴퓨터가 탑재된 로봇이 대체하기 시작하였고 단순 로봇을 뛰어넘어 지능형 로봇으로 변화되어가고 있다. 지능형 로봇 산업은 생명공학, 나노공학 등에 이어 차세대 산업으로 대두되고 있다. 지능형 로봇 산업 중에 하나의 흐름으로 연구되고 있는 것이 지능형 로봇 산업이다. 지능형로봇 산업은 큰 성장 잠재력을 가지고 있다. 로봇 산업은 첨단 신기술 분야의 복합체로 신산업의 창출을 촉진하는 로봇 컨퍼런스 연구가 활발히 이루어지고 있다. 또한 출산율 감소, 고령화 사회 진입에 따라 노동력을 대체할 수 있는 미래 산업으로 전망되고 있다.

로봇을 설계, 개발하기 위해서 실제 로봇을 사용하여 개발하면 개발비용이 높고 따라서 실패에 따른 위험부담이 크다. 이를 보완하기 위해서 범용의 목적을 지닌 확장 가능한 다양한 로봇 개발 플랫폼과 로봇 시뮬레이션 환경이 각광받고 있다. 이러한 로봇 개발 플랫폼으로 Pyro에 대해 알아보고 Pyro 플랫폼들을 비교해 본다.

Pyro는 Python Robotics의 약자로 인터프리터 언어인 Python을 사용하여 개발된 로봇 개발 플랫폼이다[1, 2, 3]. Python은 이미 교육용, 개발용으로 많이 사용되고 있으며 직관적이며 가독성이 좋은 소스코드를 생성해 낼 수 있다[4]. 또한 Python은 다른 언어와 합성하여 사용하는 합성 프로그래밍 환경이 우수하여 지금까지 개발된 다양한 로봇제어 API를 통합하기 쉽다. 사용자가 쉽게 배워 로봇과 인공지능 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[5].

본 논문의 구성은 다음과 같다. 2장에서는 Pyro를 사용하기 위해서 필요한 Player/Stage와 Python에 대해 알아본다. 3장에서는 Pyro 라이브러리와 구조, 장단점에 대해 알아보고, 4장에서는 Pyro 플랫폼들에 대해 비교해 보고 장단점을 알아본다.

2. 관련 연구

2.1 VMWare

VMWare는 PC에 새로운 OS를 설치하고 사용할 수 있게 해주는 프로그램이다. VMWare는 다양한 버전이 있으며 용도에 맞게 선택하여 사용할 수 있다.

VMWare는 실제 기계(real machine)에 VMWare 가상계층(VMWare Virtual Platform)을 생성한다. 생성된 가상계층에 가상 머신(virtual machine)을 설치하여 사용자가 설치하고자 하는 운영체제를 설치한다. 이런 방법으로 하나의 실제 기계에 다수 개의 OS가 구동될 수 있는 환경이 만들어 진다. 본 논문에서 VMWare는 두 가지 목적을

가지고 채택되었다. 첫 번째 목적은 Player/Stage라는 검증된 플랫폼을 사용하기 위해서이다. 현재 Player/Stage는 Linux/Unix 환경에서만 구동된다. 두 번째 목적은 시뮬레이션 환경을 쉽게 설치하기 위해서이다.

2.1 Player/Stage

Player는 1999년부터 지속적으로 개발된 로봇 및 센서 제어 소프트웨어이다. Player는 TCP/IP와 802.11 무선 이더넷 기반으로 분산 로봇 제어가 가능한 로봇 제어 소프트웨어 개발 프로젝트이다. Stage는 1998년부터 개발된 2차원 로봇 시뮬레이션 환경이다. Stage는 Player를 사용하여 이동 로봇 개발자가 자주 사용하는 로봇과 센서, 주변 환경들에 대하여 시뮬레이션을 할 수 있도록 개발된 시뮬레이터 소프트웨어이다. Gazebo는 2002년부터 개발된 3차원 로봇 시뮬레이션 환경이다. Stage처럼 표준 로봇 센서를 시뮬레이션 디바이스로 구현하였으며 다수 로봇을 제어할 수 있다. Stage 시뮬레이터와 고저를 사용하여 2차원보다 세밀하게 시뮬레이션이 가능한 Gazebo 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.

2.2 Python

Pytton은 1989년 Guido Van Rossum에 의해 만들어진 인터프리터 언어로, 리스트, 문자열 등의 처리가 쉽고, 동적으로 자료형을 결정하며, 큰 수를 다루는 내장 자료형이 있고, 메모리 재사용 시스템을 사용하여 메모리를 자동으로 관리해준다. 많이 사용하는 자료 구조와 알고리즘을 내장 함수와 자료형으로 구현하여 개발자로 하여금 개발 시간을 단축하고 개발 프로그램의 알고리즘 및 로직에 집중할 수 있다.

Python은 아직까지 국내에 널리 사용되지는 않지만, 그 편리함과 용이성 때문에 교육적 목적 및 실용적인 목적으로 널리 사용되고 있다[4]. 레드햇 리눅스의 설치 프로그램인 아나콘다, 구글이나 인포시크에서 사용되는 검색 프로그램, 야후의 많은 인터넷 서비스 프로그램, 다음 웹 검색 엔진 등이 Python으로 개발되었다. 또한 Python은 초기에 Perl 스크립트를 대체하기 위해서 작성된 만큼 리눅스의 스크립트 언어를 대체할 수 있으며 파일의 복사와 삭제 같은 OS 관련 기능도 충실하게 구현되어 있다. 이러한 기능을 사용하여 로봇 제어 프로그램을 테스트 하는 과정을 좀 더 편리하게 구축할 수 있었다.

3. 로봇 개발 플랫폼 Pyro

Pyro는 Python Robotics를 나타낸다. Pyro는 Python으로 제작된 로봇 개발 플랫폼이다. Pyro는 여러 대학에서 커리큘럼을 개설하여 교육하고 있는 교육적 목적과 실제 로봇에 적용하여 개발 가능한 개발 목적에 동시에 부합하는 로봇 개발 플랫폼이다[2].

Pyro는 제네틱 알고리즘과 제네틱 프로그래밍을 포함한 진보한 모듈을 가지고 있다. 예를 들면 Vision module은 널리 사용되는 비전 알고리즘과 필터를 라이브러리화하여 포함하고 있다. 또한 모든 라이브러리는 오픈소스이며 잘 문서화되어 있다. 또한 여러 모듈로 나뉘어져 있기 때문에 개발자는 다양한 커리큘럼에 맞춰 조정할 수 있다.

Pyro는 로봇을 높은 수준으로 추상화하여 프로그램으로 디자인하였다. 뿐만 아니라 다양한 프로그램을 작성할 때 다양한 플랫폼에서 코드를 전혀 수정하지 않고 동작할 수 있도록 개발하였다.

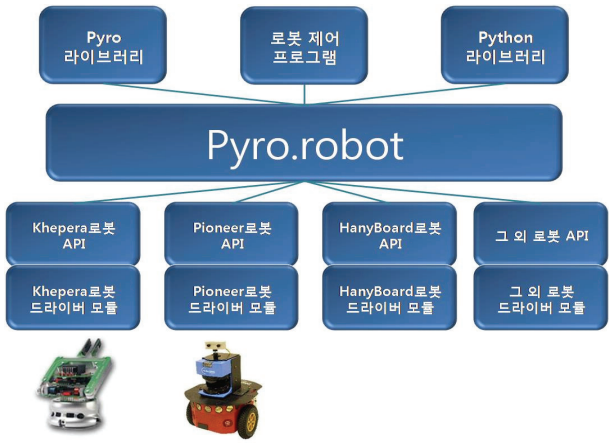
Pyro가 최소 비용으로 로봇을 배우는 효율적인 도구라는 목적을 달성하였는지 시험하는 좋은 방법은 실제로 교육이나 개발에 적용하여 보는 것이다. Swarthmore 대학에서 인공지능 교과목에 Pyro를 적용하여 좋은 반응을 얻은바 있으며 학생들은 Python과 Pyro를 매우 좋게 평가하였다.

3.1 Pyro 라이브러리

Pyro 라이브러리는 다양한 모듈을 포함한다. 각각의 모듈은 로봇제어 패러다임, 로봇 교육, 로봇 시각, 지역화, 매핑, 다중 에이전트 제어기 등으로 구성되어 있다. 로봇 제어 패러다임은 직접적/반응적/무상태 제어와 행위 기초 제어, FSM(Finite State Machine), 하부 구조, 퍼지 로직 등이다. 각 모듈은 ANNs(Artifical Neural Networks)에서 다양한 예제를 제공한다.

3.2 Pyro 구조

Pyro는 Python 클래스들의 하부 세부사항을 캡슐화 하여 만들어졌다. 그림 1은 Pyro 구조를 나타낸다[5]. 사용자는 로봇 제어 프로그램을 API로서 작성한다. 예를 들면 그림에서는 로봇들은 pyro.robot로 추상화된다. 로봇 이외에 다른 부분들은 Pyro 라이브러리로 추상화되어 사용할 수 있다. 이러한 구조는 로봇의 기능(feature)을 간단하게 하고, 세부사항, 하드웨어, 시뮬레이션 환경에 대해서 모르는 사람도 충분히 사용할 수 있도록 돕는다.



(그림) 1 Pyro 구조

3.3 Pyro 장단점

Pyro는 다양한 장점을 가지고 있다. 로봇 하드웨어에 대한 세부적인 지식이 없더라도 프로그램을 작성하고 인공지능을 테스트할 수 있도록 지원하고 있다. 또한 로봇 하드웨어를 추상화하여 어떤 로봇을 사용하더라도 개발자가 작성한 로봇 제어 프로그램을 수정하지 않고 구동시킬 수 있다. 또한 오픈 소스 프로그램으로 무료로 제공되어 실제로 로봇을 개발할 경우 개발 비용을 줄일 수 있다는 장점이 있다.

Pyro는 Python을 이용하여 각종 로봇을 시뮬레이션 하기 위한 하나의 통합 인터페이스이다. Python이라는 상위 언어를 사용하여 사용자에게 친숙하게 다가 갈 수 있는 장점이 있으며, 브레인, 엔진, 로봇, 시뮬레이터를 분리하여 효율적인 프로그래밍이 가능하며 플랫폼에 자유로운 장점을 실현하려고 한다. 현재 Pyro의 장단점과 문제점을 다음 4장에서 Pyro 플랫폼을 비교해보면서 더 자세히 알아본다.

4. Pyro 플랫폼 비교

Pyro 플랫폼은 Windows, Max OS X, Linux환경과 LiveCD가 있다. 여러 가지 플랫폼을 비교해 보고 각각의 문제점과 장단점에 대해서 알아본다.

4.1 Windows 환경

Pyro는 Windows 환경에서 win32 버전으로 설치를 하면 사용할 수 있다. Windows 환경에서 Pyro를 구동하려면 Python 2.4.1 이상의 버전이 필요하다. Pyro 구동에 필요한 Python 모듈로는 python-imaging, python-numeric이 있다. Python 모듈과 Pyro을 다운로드할 때에는 자신의 컴퓨터에 설치된 Python 버전과 모듈에서 지원하는 Python 버전이 일치해야 한다.

Windows 환경에서 Pyro를 사용하여 시뮬레이션을 하기에는 문제가 있다. Linux 기반인 Player/Stage와 연동이 되지 않아 기본적이고 간단한 시뮬레이션만 할 수 있고, 더 심화하여 수준이 높은 시뮬레이션을 할 수가 없다.

4.2 Linux 환경

현재 Pyro는 완벽하게 플랫폼 독립적으로 구현되어 있어 Windows, Mac OS X, Linux 환경에서 사용이 가능하지만, 좀 더 심화하여 시뮬레이션 하기 위해서는 Linux 환경에서 사용해야 한다. 이는 로봇에 포함되는 Real-Time OS를 Linux로 선택하여 라이센스 문제를 해결하기 위해서인 것으로 추정되는데 Player/Stage/Gazebo 등의 시뮬레이터 인터페이스는 Linux 기반으로만 구현되어 있다. 본 논문에서 기술하는 Player/Stage와 Pyro는 Debian Linux 환경에서 설치한 것을 바탕으로 작성한 것이다.

Player/Stage는 독립적으로 구동되는 로봇 제어 및 시뮬레이션 환경이나 여러 GNU 공개 라이브러리를 이용하기 때문에 사전에 설치해야하는 라이브러리가 존재한다. 필요한 라이브러리가 설치되어 있지 않으면 의존성 문제 때문에 설치가 되지 않거나 일부 기능을 사용할 수 없다. 또한 버전에 따라서도 문제가 발생할 수 있기 때문에 주의해야 한다. 사전에 설치해야할 필수적인 라이브러리는 표 1과 같다.

<표 1> Player/Stage 사전 설치 라이브러리

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

Pyro도 Player/Stage와 마찬가지로 사전에 설치해야 하는 라이브러리들이 존재한다. 예를 들어 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 Python OpenCV도 필요하다. Player/Stage와 유사하게 일부 라이브러리는 설치하지 않아도 설치가 되지만 기능이 제한된다. 예를 들어 이미지 관련 라이브러리를 설치하지 않으면 카메라 시뮬레이션이 불가능하며, Python OpenCV를 설치하지 않으면 비전 기능을 이용할 수 없다. 보다 완벽한 시뮬레이션을 하기 위해서는 표 2와 같은 라이브러리가 필요하다.

<표 2> Pyro 사전 설치 라이브러리

이름	설명
GNOME XML	XML 처리
JPEG runtime	JPEG 처리
Python OpenCV	Python용 OpenCV
TCL/TK를 위한 확장 이미지 형식 지원	Tkinter용 이미지 확장
Python 이미징 라이브러리 ImageTK	Python Tkinter용 이미지 처리

Pyro를 컴파일하기 전에 먼저 configure.py 파일을 실행하여 Pyro를 컴파일 하기 위한 옵션 설정을 한다. Pyro와 Stage를 병행 사용하기 위해서는 Stage 비전 지원 부분과, 카메라 장비를 시뮬레이션하기 위해서 카메라 관련 부분을 설정해야 한다.

Pyro의 주요 옵션으로는 영상처리 지원, 동영상 기록 지원, 비전을 이용한 Stage 시뮬레이션 지원, Aibo 비전 라이브러리 사용, 비전을 이용한 Gazebo 시뮬레이션 지원, 로봇을 이용한 지도 제작 시뮬레이션 기능 지원, 클러스터 분석도구 지원이 있다. 주요 옵션 중 마지막의 클러스터 분석도구 지원은 사용하지 않음으로 설정해야 한다. Pyro 최신 버전인 5.0.0 버전은 클러스터 분석도구가 아직 개발이 완료되지 않아 컴파일이 실패하게 된다.

4.3 LiveCD

<표 3> Pyro 플랫폼 비교

	Windows	배포판 Linux	LiveCD
Python 버전	Python 2.4 버전 이상	Python 2.4 버전 이상	Python 2.4 버전 이상
설치해야 할 라이브러리	python-imaging python-numeric	표 1, 표 2 참조	·
사용 가능한 시뮬레이터	Pyro	Pyro, Player, Stage, Gazebo	Pyro, Player, Stage
한글 지원	한글 지원 가능	한글 지원 가능	한글 지원 불가능
파일 저장 지원	지역파일 시스템에 저장 가능	지역파일 시스템에 저장 가능	별도 마운트 과정을 거쳐야 함

Pyro는 위와 같은 복잡한 설치과정을 최대한 단순화하기 위해서 LiveCD를 제공하고 있다. LiveCD란 CD로 부팅이 가능하며 램 드라이브를 하드디스크로 사용하는 형태의 운영체제이다. LiveCD는 운영체제를 설치할 필요가 없으며 종류에 따라 여러 응용프로그램이 탑재되어 있다. Pyro는 Knoppix 기반으로 제작된 LiveCD를 배포하고 있으며 부팅을 마치면 Pyro/Player/Stage를 사용할 수 있다. Gazebo는 아직 하드웨어 호환성 문제로 모든 PC에서 완벽하게 사용할 수 없기 때문에 LiveCD에는 빠져있다. 또한 영문 Knoppix 기반으로 제작되어 있기 때문에 한글을 사용할 수 없는 단점이 있다. 그리고 사용하면서 직접 작성한 코드나 파일을 저장할 수 없어서 ftp로 다른 서버가 필요하거나 하드 디스크를 마운트 시켜서 사용해야 하는 단점이 있다.

5. 결론

본 논문에서는 로봇 개발 플랫폼인 Pyro에 대해 알아보고 Pyro 플랫폼들에 대해 표 3과 같이 비교해 보았다. Pyro는 Python이라는 언어를 사용하여 사용자에게 친숙하게 다가갈 수 있고 오픈 소스 프로그램으로 무료로 제공되어 실제로 로봇을 개발할 경우 개발 비용을 줄일 수 있다. 로봇 하드웨어에 대한 세부적인 지식이 없더라도 프로그램을 작성하고 인공지능을 테스트할 수 있도록 개발되었다. 또한 로봇 하드웨어를 추상화하여 어떤 로봇을 사용하더라도 개발자가 작성한 로봇 제어 프로그램을 수정하지 않고 구동시킬 수 있다. 로봇 개발에 실제 로봇을 사용할 경우 시간과 비용이 많이 들어 개발에 실패하거나 문제가 생겼을 경우에 위험부담이 크다. 그러나 위험부담을 줄이고 개발기간을 단축하기 위해서 실제 구성될 환경과 동작 환경을 고려한 시뮬레이션 환경으로 Pyro는 아주 적합한 로봇 개발 플랫폼이다.

Pyro는 LiveCD로 사용할 수 있고, Windows, Mac OS X, Linux 환경에서 사용이 가능하지만, 좀 더 심화하여 시뮬레이션하기 위해서는 Linux 환경에서 사용하는 것이 가장 좋다. 하지만 Linux 환경에서 Pyro를 사용하려면 여러 가지 필요한 라이브러리와 패키지 의존성 문제도 고려해야 한다. 이런 점들을 잘 고려하여 Pyro를 사용한다면 국내 로봇기술 산업의 발전을 도모할 수 있을 것이다.

실제로 로봇 공학에 지식이 없는 학생들을 대상으로 Pyro를 이용하여 교육을 해보았는데 비교적 무난하게 인공지능 프로그램을 작성하는 것을 볼 수 있었다. 로봇 개발이 쉽고, 가시적인 결과확인이 가능한 시뮬레이션 환경을 사용하여 학생들의 자발적인 참여로 로봇 공학 및 인공지능 교육에 효율적인 플랫폼임을 확인하였다.

참고문헌

[1] Douglas Blank, Deepak Kumar, Lisa Meeden, and Holly Yanco. “Pyro: An intergrated environment for robotics education”, In AAAI-05 Mobile Robot Program, pages 1718-1719, 2005

[2] Douglas Blank, Deepak Kumar, Lisa Meeden and Holly Yanco. “Pyro: A Python-based Versatile Programming Environment for Teaching Robotics”, In Journal of Educational Resources in Computing, 2003

[3] Douglas Blank, Lisa Meeden, and Deepak Kumar. “Python robotics: an environment for exploring robotics beyond legos”, In SIGCSE ’03: Proceedings of the 34th SIGCSE technical symposium on Computer science education, pages 317 - 321, 2003

[4] Jeffrey Elkner, “Using Python in a High School Computer Science Program”, 8th International Python Conference, 2002

[5] Douglas Blank, Deepak Kumar, Lisa Meeden, Holly Yanco, “The Pyro toolkit for AI and robotics”, To appear in AI Magazine, 2006

[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, 2003