

<http://dx.doi.org/10.7236/JIIBC.2013.13.2.165>

JIIBC 2013-2-22

기능 축약화 스크립트를 활용한 로봇 시뮬레이션 저작 도구

Authoring Tool for Robot Simulation Creation using Functional Concise Script

김영준*, 서용호**

Young-Joon Kim, Yong-Ho Seo

요약 본 논문은 기능 축약화 스크립트를 이용한 로봇 시뮬레이션을 제작하는 방법을 제안한다. 일반적인 로봇 시뮬레이션 저작 도구들은 이를 사용하기 위한 전문적인 개발 언어와 이와 연동되는 API 등에 대한 전문적인 지식을 요구하고 있으며, 이러한 사전지식들은 로봇 시뮬레이션을 환경을 구현하는 데 매우 큰 장애 요인이 되고 있다. 이러한 문제를 해결하기 위해서 본 논문은 시뮬레이션 환경을 구성하기 위해 기능 축약화 스크립트를 설계하였으며, 축약화된 스크립트는 단순화된 서비스 명령들과 최소의 옵션들로 구성된다. 실험에서는 개발된 저작 도구를 고등학교 학생과 선생님들에게 교육하였으며, 초보자들이 본 저작도구 하루의 교육과정 이수를 통해 로봇 시뮬레이션 환경을 구성하고 시뮬레이션 환경의 로봇을 제어할 수 있음을 검증하였다.

Abstract This paper proposes a method of simplification for robot simulation creation using functional concise script. General robotics simulation authoring tools require professional knowledge regarding development language and their APIs, and these prerequisites are considered as a big obstacle in implementing robotics simulation. In order to resolve these problems, we developed an authoring tool using functional concise script in building simulation environment, and this concise script composes of simplified service commands and minimum options. In experiment, we educated students and teachers in High school to use this authoring tool and we verified that the proposed method makes it possible to build robotics simulation environment and control simulated robots through one-day training course for the novices.

Key Words : Robotics Simulation, Functional Concise Script, Mobile Robots

1. 서 론

로봇 시뮬레이션은 로봇 제품의 개발 및 로봇틱스 연구에 있어서 필수적인 요소로서 자리매김하고 있으며, 실제 하드웨어 로봇을 사용하기 어려운 경우이거나 하

드웨어 제작 전 단계에서 구현하고자 하는 기능의 테스트 및 실험 데이터 수집을 위해 많은 분야에서 사용되고 있다^[1].

최근에는 로봇 관련 전공자뿐만 아니라 일반인들이나 학생들에 있어서도 로봇에 대한 관심이 높아지면

*준회원, 목원대학교 지능로봇공학과

**중신회원, 목원대학교 지능로봇공학과 (교신저자)

접수일자 2013년 1월 28일, 수정완료 2013년 3월 8일

게재확정일자 2013년 4월 12일

Received: 28 January 2013 / Revised: 8 March 2013 /

Accepted: 12 April 2013

**Corresponding Author: yhseo@mokwon.ac.kr

Dept. of Intelligent Robot Eng., Mokwon University, Korea

서, 시뮬레이션을 활용한 로보틱스 원리학습과 창의력 배양, IT 기술 실습 등 다양한 목적으로 로봇 시뮬레이션을 활용하고자 하는 시도가 점점 더 증가되고 있는 상황이나, 이에 대한 활용 측면을 살펴보면 대부분의 사용자는 이미 만들어진 시뮬레이션 환경 속에서 사전에 미리 정의되어 있는 시뮬레이션 로봇을 제어하는 기능만 수행할 수 있을 뿐, 사용자가 원하는 형태의 시뮬레이션 로봇을 제작하거나 환경을 구성하는 데에는 상당한 노력과 전문 지식을 요구하고 있어, 현실적으로 로봇 시뮬레이션의 보급화에 많은 걸림돌이 되어 왔던 것이 주지의 사실이다^[2].

특히, 로봇 시뮬레이션 분야는 단순히 컴퓨터 개발 언어와 로보틱스에 대한 지식만을 필요로 하는 것이 아니라 3D 가상현실 상에서 렌더링 엔진 및 물리엔진 등과 같은 전문적인 기술분야와 3D 모델링과 같은 그래픽 분야의 기술 등이 사용되기 때문에 로봇분야의 전문가에게도 접근이 힘든 분야로 여겨져 왔다.

본 연구에서는 로봇 시뮬레이션을 구성하는데 있어서 이러한 현실적인 문제를 해결하고자, 초보자가 별도의 교육과정 없이 쉽게 로봇 시뮬레이션 환경과 시뮬레이션 로봇을 제작할 수 있도록 하는 기능이 축약화된 스크립트 기반의 실행 환경을 제안하며, 제안된 연구결과를 활용하였을 경우, 로봇 분야에 대한 사전 지식이 없는 초급자의 경우에도 1일 과정의 실습을 통하여, 원하는 형태의 시뮬레이션 로봇을 직접 제작할 수 있음을 확인할 수 있었다.

II. 로봇 시뮬레이션 개발 과정

표 1은 대표적인 로보틱스 시뮬레이션 제품들과 해당 제품들에서 지원하고 있는 개발 언어를 나열한 것으로, 대부분의 툴들이 C/C++ 그리고 C#과 같은 전문적인 개발 언어를 지원하고 있는 것을 확인할 수 있다^[3].

로봇 시뮬레이션은 그림 1과 같이 마이크로소프트의 로보틱스 개발자 스튜디오(MS Robotics Developer Studio: MSRDS)를 비롯한 상용 제품들과 Gazebo와 같은 오픈 소스 형태의 제품 등 여러 가지 형태의 제품들이 존재하며, 각각의 제품들은 자체적인 개발 언어를 지원하고 있다^[4-5].

C, C++ 및 C#, Java 등과 같은 전문적인 개발용 프로그래밍 언어는 개별 학습 훈련만 최소 3개월 이상이 소

요되며, 대학에서도 한 학기 과정이 소요될 만큼 많은 시간이 요구되고 있으며, 이로 인해 전공자가 아닌 일반인이나 비전공 학생들의 경우에는 사실상 로보틱스 시뮬레이션을 접할 수 있는 기회가 거의 제공되지 않고 있고, 또한 관련 교육과정도 쉽게 개설되지 못하고 있는 것이 사실이다.

표 1. 로보틱스 시뮬레이션 툴과 사용되는 개발 언어
Table 1. Robotics Simulation Tools and Development Languages

Prod. Name	Prod. Type	Language
Gazebo	Open Source	C++
MSRDS	Commercial	C#
Virtual Robot Experimentation Platform	Commercial	C/C++
AnyKode Marilou	Commercial	C/C++, C++ CLI, C#, J#
Webots	Commercial	C/C++, Java, Python, URBI
RoboticsLab	Commercial	C++
EDRS	Commercial	C/C++, Flow



그림 1. MSRDS 시뮬레이션 실행 예
Fig. 1. Example of MSRDS Simulation

본 연구에서는 이러한 문제를 해결하고 비전문가 또는 일반인이 사전 교육없이 쉽게 로봇 시뮬레이션 환경을 구성하여 시뮬레이션 로봇을 제어할 수 있도록 하는 기능 집약화 스크립트 기반의 시뮬레이션 환경 구성 방법을 제안하였다. 또한 제안된 방법을 마이크로소프트의 로보틱스 개발자 스튜디오의 시뮬레이션 환경에서 구동이 가능하도록 구현하였다. 실험에서는 구현된 로봇 시뮬레이션 저작도구를 이용해 고등학교 학생 및 교사를 대상으로 실습을 진행하였으며, 그 결과를 토대로

제안된 방법에 대한 효과를 제시하였다.

III. 기능 축약화 스크립트

일반적으로 로봇 시뮬레이션은 매우 많은 수의 명령과 단계, 그리고 복잡한 옵션들로 구성된다. 이러한 복잡한 명령어들과 옵션들을 기존의 개발 언어에서 지원하는 방식으로 구현한다면 상당한 양의 코드가 필요로 하게 되고 개발자가 일일이 해당 명령어와 옵션들의 의미를 파악하고 있어야 하는 문제 또한 제기된다.

본 연구에서는 많은 수의 명령어 수를 줄이고 복잡한 옵션의 사용을 최소화 하기 위해 필요한 명령어와 옵션들을 Off-The-Shelf 형식의 서비스로 설계하였으며, 각 서비스는 그림 2와 같이 해당 서비스 명령어와 옵션으로만 구성되도록 단순화 하였다.

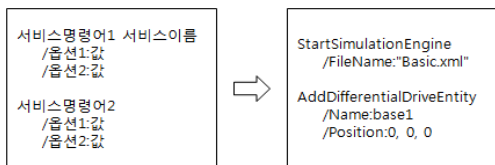


그림 2. 기능 축약화된 서비스 명령어 예
Fig. 2. Example of Service Commands

기능 축약화의 핵심은 가장 빈도가 높은 옵션 값들을 내부적으로 기본 값으로 설정한 후, 사용자가 최소한의 옵션만 변경하여 해당 서비스 명령어들을 사용하도록 하는 것으로서, 기능들을 서비스 내부적으로 축약시키되, 외부적으로 공개되는 옵션들을 그림 3과 같이 최소화 시키는 것이다.

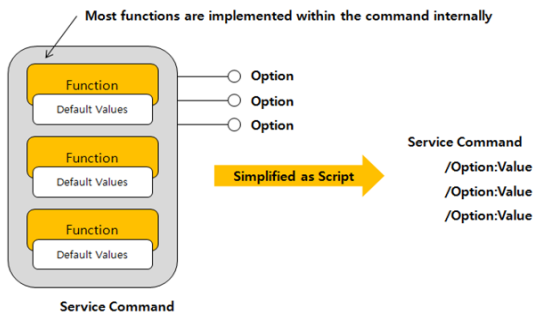


그림 3. 기능 축약화 과정
Fig. 3. Deriving Function Concise Command

이러한 서비스 명령어의 정의는 본 연구에서 정의형 스크립트(Definitional Script)로 불리며, 특별히 시뮬레이션 환경 구성과 로봇 설계 시에 사용되도록 하였고, 컴퓨터의 OS에서 사용되는 일반적인 시스템 명령어의 사용방법을 따름으로서, 개발자가 쉽게 사용 방법을 익히도록 고려하였다.

또한 정의형 스크립트 외에 일반적인 개발 언어에서와 유사한 기능을 제공하기 위해 제어형 스크립트(Control Script) 기능을 정의하였으며, 흐름 제어 및 반복문, 함수 정의 등으로 구성되고 그림 4와 같이 JavaScript 언어의 문법을 따르도록 하였다^[6].

```

1 StartSimulationEngine
2   /FileName:"SimState/BasicObstacles.xml"
3
4
5 AddDifferentialDriveEntity base1
6   /Position:0 0 0
7
8 FlushScript
9
10 base1.GoTo(1.0, 0.5)
11
12 base1.Turn(90, 0.2)
13
14 base1.Go(0.5, 0.5)
    
```

그림 4. 모바일 로봇을 정의하기 위한 스크립트의 구성
Fig. 4. Script parts for defining mobile robot

그림 5는 기능 축약화된 스크립트의 실행과정을 순서대로 표현한 것으로서, 스크립트 명령어의 내부 실행시 정의형 명령어와 제어형 명령어를 각기 분리한 다음, 정의형 명령어에 대해서는 사전에 객체 형태로 생성만 시킨 후, 제어형 명령어의 실행 시점에 따라 실행되도록 구현하였다.

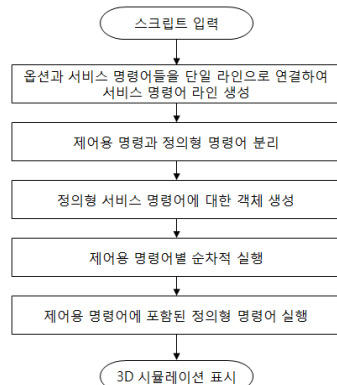


그림 5. 스크립트 실행을 위한 순서도
Fig. 5. Flow Chart for Executing Script

IV. 실험 및 결과

본 연구에서는 제안된 방식의 구현을 위하여 기능 축약화된 스크립트를 정의한 후, MSRDS R4 버전의 시뮬레이션 기반 위에서 실행되도록 구현을 하였으며, 제안된 스크립트 방식의 효율성과 접근성 성능을 확인하기 위하여, 다음 그림 6과 같은 시나리오를 구현하여 사용자가 직접 환경을 설계하고 필요한 로봇을 설계한 후, 장애물을 회피하도록 수행하였다.

- 장애물 환경 구성
 - 두 개의 장애물 박스 추가
- 모바일 로봇 정의
 - 두 개의 휠 및 LRF 센서, 범퍼 센서 장착
- 장애물 회피
 - LRF 센서 데이터를 기반으로 로봇이 장애물 사이를 반복하여 이동하도록 제어

그림 6. 실험에 사용된 로봇환경 및 시나리오
Fig. 6. Robot Environment and Scenario for Experiment

1. 시뮬레이션 환경구성 스크립트

시뮬레이션 환경을 구성하기 위해 사전에 하늘과 바닥, 태양 등이 미리 정의되어 있는 기본 환경으로부터 시뮬레이션 엔진이 구동되도록 하였으며, 장애물 기능을 구현하기 위해 두 개의 기다란 박스를 그림 7과 같이 추가하였다.



그림 7. 환경 구성을 위한 스크립트
Fig. 7. Script for Definition of Environment

2. 모바일 로봇 구성 스크립트

모바일 로봇은 두 개의 휠과 범퍼 센서, 그리고 하나의 레이저 거리 센서를 장착한 로봇으로 정의하였으며, 3D 모델링은 파이어나사사의 P3DX 형태를 그림 8과 같이 사용하였다.

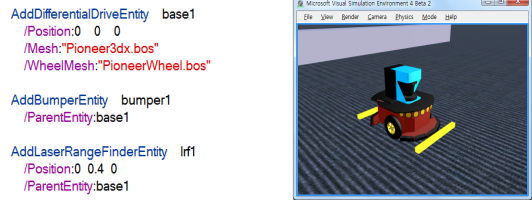


그림 8. 로봇의 3D 메시가 추가된 스크립트
Fig. 8. Script for Definition of 3D Modeling Mesh

3. 센서 활용 스크립트

시뮬레이션에서 가상의 센서를 이용해 장애물과의 거리를 감지한 후, 일정 거리 이내로 접근하였을 경우에는, 로봇을 180도 회전시켜 다시 전진을 하도록 스크립트를 그림 9와 같이 구현하였으며, 장애물과의 거리를 측정하기 위해 시뮬레이션 환경의 로봇에는 LRF(Laser Range Finder) 센서를 이용하였다.

```

FlushScript

base1.Go(0.4, 0.4)

while (true)
{
    distances = lrf1.Get()
    d180 = distances.DistanceMeasurements[180]

    if (d180 < 1000)
    {
        base1.Turn(180, 0.3)
        base1.Go(0.4, 0.4)
    }
}
    
```

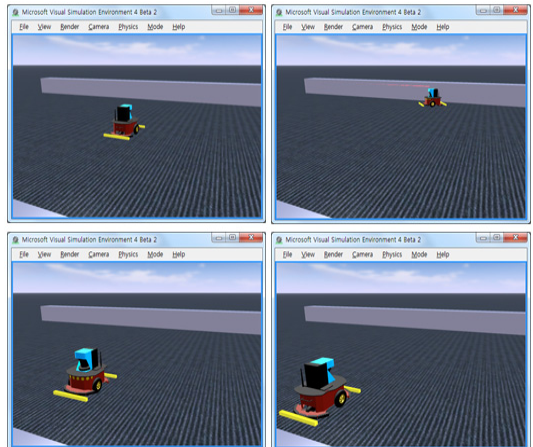


그림 9. 장애물 회피를 위한 로봇 제어 스크립트(상) 및 장애물 회피 스크립트 실행결과(하)
Fig. 9. Robot Control Script for Obstacle Avoidance (Upper) and Execution Result of script(Lower)

4. 초보자를 대상으로 한 교육과정 비교 실험

표 2는 일반 대학에서 다루는 학과목의 소요 시간을 기준으로 로봇 시뮬레이션에 필요한 교과목과 시간을 산정한 것으로서, 초보자를 대상으로 할 경우, 약 10개월 정도 또는 최소 한 학기 이상의 시간이 소요되는 것을 알 수 있다.

반면, 표 3에서와 같이 마이스터고 교사 및 카이스트 IT영재학교 고등학생들에게 적용한 교육과정에서는 로봇 시뮬레이션 교육에 총 8시간 정도가 소요됨을 알 수 있다.

표 2. 로보틱스 시뮬레이션 교육에 필요한 교과 항목
Table 2. Prerequisites for Robotics Simulation

Subjects	Time
Development Tool Training	3개월
Basic Robotics Knowledge Training	1개월
Robotics Simulation API or SDK Training	3개월
3D Modeling Training	3개월

표 3. 제안된 방식을 고등학교 교사 연수에 적용한 경우
Table 3. Curriculum of Proposed Method

Subjects	Time
Script Introduction	30분
Simulation Environment	1시간
Entity Design	1시간
Robot Design	1시간
Robot Control	2시간
Sensor Control	2시간

V. 결론

본 논문에서는 사용자가 원하는 형태의 시뮬레이션 로봇을 제작하거나 환경을 구성하는 데에 상당한 노력과 전문 지식을 요구하고 있어, 현실적으로 로봇 시뮬레이션의 보급화에 많은 걸림돌이 되어 왔던 문제를 해결

하고자, 초보자가 별도의 교육과정 없이 쉽게 로봇 시뮬레이션 환경과 시뮬레이션 로봇을 제작할 수 있도록 하는 기능이 축약화된 스크립트 기반의 실행 환경을 제안하였으며, 제안된 연구결과를 활용하였을 경우, 로봇 분야에 대한 사전 지식이 없는 초급자의 경우에도 1일 과정의 실습을 통하여, 원하는 형태의 시뮬레이션 로봇을 제작할 수 있음을 확인할 수 있었다.

향후 연구과제로는 제안된 방법에 사용되는 스크립트를 스마트폰에서도 실행되도록 하여, PC 없이도 스마트폰만을 이용하여, 로봇 시뮬레이션이 가능하도록 연구하는 것이다.

참고 문헌

- [1] Railsback Steven F, Lytinen Steven L and Jackson Stephen K. Agent-based simulation platforms: Review and development recommendations. Simulation 2006;82(9): 609 - 623.
- [2] Y. H. Seo, "Vision Based Tracking Method for Mobile Robot Using Network Service Programming", Journal of Korean Institute of Information Technology, vol. 6, issue 6, pp. 1-7, Dec 2008.
- [3] L. Brakmo and L. Peterson, "Experiences with Network Simulation," Proc. ACM SIGMETRICS 96, ACM Press, New York, 1996, pp. 80-90
- [4] Microsoft Robotics Developer Center, <http://msdn.microsoft.com/robotics>
- [5] Oscar Almeida, Johannes Helander, Henrik Nielsen, Nishith Khantal: Connecting Sensors and Robots through the Internet by Integration Microsoft Robotics Studio and Embedded Wen Services, Proceeding of IADIS International Conference (2007)
- [6] <http://www.helloapps.com/>

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 연구임(No. 2011-0013776).

저자 소개

김 영 준(Young Joon Kim)



- 2005년 : Microsoft Robotics Group 연구원
- 2009년 ~ 현재 : (주)헬로엠플스 대표이사
- 2012년 ~ 현재 : 목원대학교 IT공학과(박사과정)

<관심분야 : 로봇 시뮬레이션, 인공지능, 스마트폰 원격제어>

서 용 호(Yong-Ho Seo)



- 1999년 : KAIST 전산학과(공학사)
- 2001년 : KAIST 전자전산학과(공학 석사)
- 2007년 : KAIST 전자전산학과(공학 박사)
- 2007년 : 미국 MS 로보틱스그룹 인턴 연구원

• 2008년 : 미국 쉐콤 과장

• 2010년 ~ 현재 : 목원대학교 지능로봇공학과 조교수

<관심분야 : 지능로봇, 로봇비전, 휴머노이드, 임베디드시스템>