

# 진화 하드웨어상에서 유전자 프로그래밍에 의한 온라인 학습

석호식, 이광주, \*이강, 장병탁  
서울대학교 컴퓨터공학과, \*한동대학교 전산전자공학부  
{hsseek, kjlee, btzhang}@scai.snu.ac.kr, \*yk@handong.edu

## On-line Learning by Genetic Programming on Evolvable Hardware

Ho-sik Seok, Kwang-Ju Lee, \*Kang Yi, and Byoung-Tak Zhang  
Dept. of Computer Engineering \*School of CS&EE  
Seoul National University Handong University

### 요 약

본 논문에서는 진화 하드웨어에 기반한 자율 이동 로봇의 온라인 학습 기법에 관하여 소개하고자 한다. 진화 하드웨어는 실행 시간중에 하드웨어 회로 구성을 변경시킬 수 있는 새로운 개념의 FPGA이다. 제어 프로그램은 진화 하드웨어상에 트리 형식으로 구현되며 유전자 프로그래밍을 이용하여 학습하게 된다. 로봇의 환경 탐사가 진행됨에 따라 입력되는 센서 정보에 기반하여 제어 프로그램은 학습을 수행하게 되며, 노드 돌연변이의 유전 연산자를 이용하여 진화한다. 제어 프로그램의 게이트 회로는 학습의 진행에 맞추어 실행 시간중에 보다 적합도가 높은 방향으로 발전한다. 본 논문에서는 진화 하드웨어를 이용한 이러한 학습 방식과 FPGA 구현 및 로봇 제어에의 응용에 대한 실험 결과 등을 설명할 것이다.

## 1. 서 론

진화 하드웨어 혹은 재구성 하드웨어 (reconfigurable hardware)는 실행 시간 중 하드웨어의 구성을 변경할 수 있는 FPGA (field programmable gate array)이다.

실행 시간 중 재구성이 가능하다는 특징은 진화 하드웨어를 결합 허용 시스템, 저전력 시스템, 적응 시스템, 회로 설계 등에 대한 새로운 접근 방법으로 제시하고 있다. [1]

진화 하드웨어는 환경의 변화에 맞추어 최적의 회로를 재구성할 수 있기 때문에, 적응 시스템에의 응용 가능성이 기대되고 있다. [2], [3]

유전자 프로그래밍을 진화 하드웨어 상에서 구현하는 것은 인자들의 미세 조정에는 불리하다. 그러나 개체에 대한 진화 연산과 적합도 평가를 병렬적으로 수행할 수 있으므로 병목 지점으로 작용하는 적합도 평가에서 소모되는 시간을 단축시킬 수 있다. [4]

본 논문에서는 진화 하드웨어를 이용한 로봇 제어를 위하여 소개한다. 로봇 제어가 진화 하드웨어 상의 회로 형태로 구성되어 있을 경우, 환경 변화에 따라 최적의 제어 회로를 구성하는 것이 가능해진다. 환경에 대한 학습을 통하여 제어 회로를 재구성할 수 있다는 것은 로봇의 활용 범위를

를 다양하게 하는 것을 가능하게 한다.

본 논문에서는 센서 입력에 대하여 최적의 반응을 찾는 로봇 제어를 구현하여 실험해 보았다. 로봇은 운동을 하며 입력되는 환경 정보와 기존의 입력 정보 해석 논리를 통해 새로운 제어 회로를 구성하는 것을 목표로 한다.

논문의 구성은 다음과 같다. 2장에서는 제어 프로그램을 설명하며, 3장은 제어 프로그램의 진화를 다룬다. 4장은 진화 하드웨어의 구현 및 실험에 대하여 설명할 것이며 5장에서는 연구의 의의와 앞으로의 연구에서 추가할 사항을 담고 있다.

## 2. 로봇 제어 프로그램

### 2.1 로봇 운동

로봇은 장애물 환경에서 광원으로 표시되는 목적지를 찾아가는 것을 목표로 한다. 로봇은 표 1의 센서 시스템을 이용하여 환경 정보를 입력받게 된다.

- |                                                                                                            |
|------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• 초음파 센서: 장애물과의 거리 탐색용 - 3개</li><li>• 광센서: 목표의 방향 탐색용 - 6개</li></ul> |
|------------------------------------------------------------------------------------------------------------|

표 1. 로봇의 센서 시스템

로봇 제어기는 그림 1과 유사한 모습의 트리 구조를 취하

고 있으며, 환경 탐사를 통하여 최적의 트리 구조를 찾아내는 것이 로봇의 목표이다.

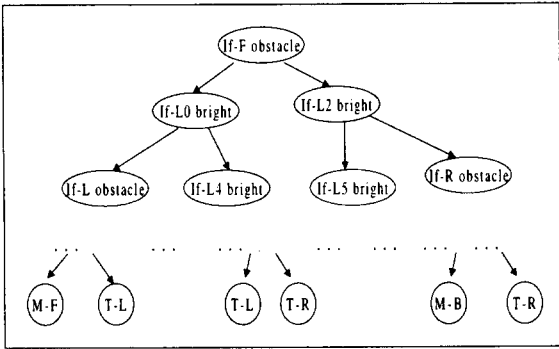


그림 1. 개체의 예

### 2.2 로봇 제어기

그림 1과 같은 모습의 개체가 로봇 제어 회로를 구성하게 된다. 개체 하나가 제어 회로를 구성하며, 한정된 시간동안 운동을 한 후 적합도를 평가받게 된다.

개체의 Terminal set과 Non-terminal set은 표2와 같이 구성된다.

<ul style="list-style-type: none"> <li>Terminal set: {                     <ul style="list-style-type: none"> <li>· MF, MB</li> <li>· MLF, MRF</li> </ul> </li> <li>Non-terminal set:                     <ul style="list-style-type: none"> <li>{                             <ul style="list-style-type: none"> <li>· if L0, if L1, if L2, if L3, if L4, if L5</li> <li>· if US0, if US1, if US2</li> </ul> </li> </ul> </li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

표 2. 개체의 노드

Terminal set은 로봇의 운동 정보를 나타낸다. 전진과 후진, 좌로 전진, 우로 전진의 네 가지 운동이 가능하다.

Non-terminal set은 환경 정보 해석을 담당한다.

if L0 ~ if L5는 6개의 광센서 입력 정보 중 어떤 센서가 가장 큰 입력값을 갖는지의 여부를 나타낸다.

if US0 ~ if US2는 3개의 초음파 센서 입력 정보를 해석하는 역할을 담당한다.

로봇은 이 두 종류의 센서 입력 정보를 이용하여 목적지의 위치와 장애물까지의 거리를 파악하게 되며 입력 정보 패턴에 대하여 어떤 운동을 취할지 결정하게 된다.

## 3. 제어 프로그램의 진화

### 3.1 진화 연산 인자

표3에 제어기 진화를 위한 진화 연산 인자들이 소개되어 있다.

개체는 깊이 8의 이진 트리 모습을 하고 있다. 트리의 깊이가 8로 한정된 것은 진화 하드웨어의 Placement & Routing 자원 제약으로 깊이 8 이상의 이진 트리 표현은 불가능하기 때문이다. 한정된 세대수 동안만 진화 연산을 수행한 후 종료시키는 것은 로봇이 실제 상황

에서 운동하는 경우를 감안한 설정이다.

<ul style="list-style-type: none"> <li>· 개체의 수: 10</li> <li>· 개체의 모습: 깊이 8의 이진 트리</li> <li>· 유전 연산: node mutation</li> <li>· 종료 정책: 한정된 세대수 동안 진화 수행</li> </ul>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------

표 3. 진화 연산 파라미터

### 3.2 진화 연산과 적합도 평가

로봇의 학습을 위하여 사용한 연산자는 돌연변이 연산이다. 이 경우 돌연변이 연산은 한 노드를 다른 노드로 변화시키는 것을 목표로 한다. 이를 위해서는 노드를 구성하고 있는 회로를 다른 회로로 변경시켜야 하는데 이는 진화 하드웨어의 제어 프로그램에서 처리하게 된다.

적합도 평가는 식 (1)과 같이 표현된다.

$$F_{new} = F_{old} + 1/\lambda - I(U) \cdot \text{penalty} \quad \dots(1)$$

$$I(U) = \begin{cases} 1 & \text{장애물과의 거리가 일정값 이하일때} \\ 0 & \text{그 이외의 경우} \end{cases}$$

· λ: 광원과의 거리

적합도는 운동을 시작해서 운동을 종료하기까지 총 움직임을 통하여 계산된다. 로봇과 장애물까지의 거리가 일정 한도이하가 되면 페널티가 주어지게 되며 광원까지의 거리가 줄어들수록 큰 적합도 값을 갖게 된다.

## 4. 제어 프로그램의 하드웨어 구현 및 실험

### 4.1 개체의 표현과 진화

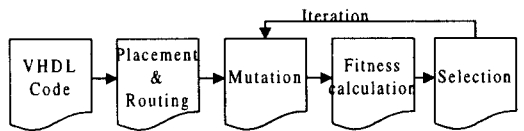


그림 2. 하드웨어 상에서의 진화

그림2에 하드웨어상에서의 진화 과정이 대략적으로 보여지고 있다. VHDL (VHSIC Hardware Description Language)을 이용하여 회로를 설계한 후 이 회로를 placement & routing하게 된다. 진화 하드웨어에 올려진 제어 회로는 돌연변이를 통하여 트리 구조를 변형시키게 되며, 10개의 개체에 대하여 운동을 반복한 후 가장 높은 적합도를 갖는 개체를 찾게 된다. 최고의 적합도를 갖는 개체의 정보를 이용하여 진화 연산을 실시한 후 그림 2의 과정을 반복하게 된다.

그림 3에서 개체의 하드웨어 표현 모습을 보이고 있다. 그림 3과 같은 모습의 노드들이 개체를 구성하게 된다. 각 노드에서는 9개의 입력 정보를 이용하여 하위 경로중 어느 경로를 참으로 할 것인지 결정하게 된다.

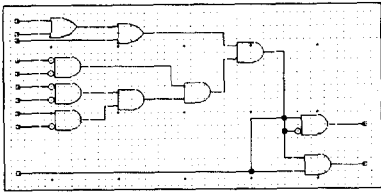


그림 3. 노드의 표현 예

진화 연산은 노드를 구성하고 있는 게이트를 변경하여 새로운 회로를 구성하는 방식으로 이루어 진다. 돌연변이 과정에서 특기할 점은 회로의 변경이 실행 시간 중에 이루어진다는 것이다. VHDL 코드 작성시 RECONFIG 속성을 부여하여 코드를 작성하면 Dynamic reconfiguration될 게이트들의 정보가 배열 형태로 저장되며, 이는 제어 프로그램에 의하여 하드웨어 오브젝트의 형태로 진화 하드웨어에 올려지게 된다. 이 과정을 통하여 실행 시간 재구성성이 가능하게 된다.

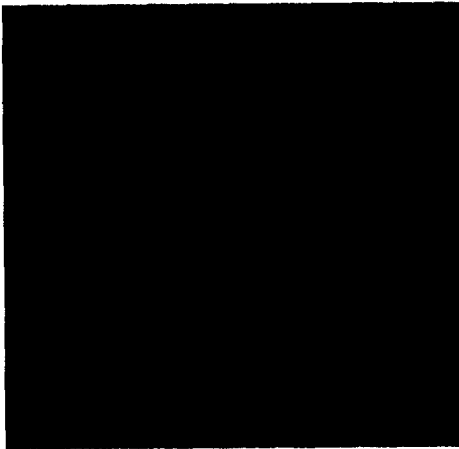


그림 4. Placement&routing된 개체 모습

그림 4에서는 진화 하드웨어에 최종적으로 표현된 개체의 모습을 보이고 있다. 그림 4에서 직사각형 형태를 하고 있는 블록이 트리의 노드이며 나머지 부분은 Routing에 이용되었음을 알 수 있다.

#### 4.2 실험 결과

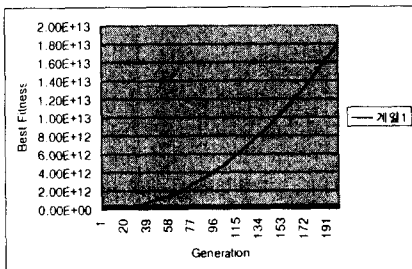


그림 5. 적합도 변화 그래프

그림 5에 최고 적합도를 갖는 개체의 적합도 변화를 보이고 있다. 적합도의 증가를 통하여 개체의 학습이 이루어지고 있음을 파악할 수 있다.

#### 5. 결론

본 논문에서는 진화 하드웨어에 기반한 로봇 제어를 구현하였다. 진화 하드웨어에 기반하여 회로를 구현할 경우 다음과 같은 장점을 갖게 된다. 첫째, 수행 작업에 특화된 회로를 구현할 수 있다. 둘째, 계산의 병렬화를 통하여 계산 속도를 향상시키며, 셋째, 작업 경과와 함께 학습을 진행시켜 보다 적합도가 높은 회로를 구성하는 것이 가능하다.

향후 본 논문을 발전시켜 다음과 같은 사항을 연구하여야 할 것이다. 첫째, 개체를 표현함에 있어 칩의 자원 이용량이 보다 적은 표현형을 찾아야 한다. 둘째, 학습 결과를 표현함에 있어 하드웨어에 적합한 모습으로 자동으로 변환하여 주는 방법론을 개발하여야 할 것이다.

감사의 글: 본 연구는 과학재단 핵심전문 연구(과제 번호 981-0920-107-2)에 의해 지원되었음.

#### 참고 문헌

- [1] Paul Layzell, The 'Evolvable Motherboard' A Test Platform for the Research of Intrinsic Hardware Evolution, *Cognitive Science Research Paper 479*, 1998.
- [2] Weixin Liu et al, ATM Cell Scheduling by Function Level Evolvable Hardware, *International Conference on Evolvable Systems 1996*, pp. 180-192, 1996.
- [3] Isamu Kajitani, et al, A Gate-Level EHW Chip: Implementing GA Operations and Reconfigurable Hardware on a Single LSI, *International Conference on Evolvable Systems 1998*, pp. 1-12, 1998.
- [4] Koza, John et al, Evolving computer programs using rapidly reconfigurable field programmable gate arrays and genetic programming, *Proceedings of the ACM Sixth International Symposium on Field Programmable Gate Arrays*. New York, NY: ACM Press. pp. 209-219, 1998.