
멀티-도메인 공학시스템의 자동설계방법

서기성* · 박세현**

Automated Design Method for Multi-domain Engineering Systems

Kisung Seo* · Se-hyun Park**

본 연구는 2002년도 한국과학재단 목적기초연구 R01-2002-000-00588-0 지원으로 수행되었으며, 이에 감사드립니다.

요 약

멀티-도메인 공학시스템은 전기, 기계, 유압, 열등의 도메인을 포함하며, 시스템 구성이 복잡하여 설계에 많은 어려움을 가지고 있다. 최적의 설계를 위해서는 각 도메인에 대한 통합된 설계 방법과 자동적이고 효율적인 탐색방법이 요구된다. 본 논문은 도메인에 독립적인 본드 그래프(bond graph)와 대규모 공간 해의 탐색에 적합한 진화 알고리즘의 일종인 Genetic Programming(유전 프로그래밍, GP)를 결합하여 멀티 도메인 동적시스템에 대한 디자인 해를 자동적으로 생성해주는 설계 방법을 제시하였다. 제안된 설계방법의 효용성을 입증하기 위해서 고유값(eigenvalue) 설계 문제가 실험되었고, 서로 다른 태아모형을 가진 고유값의 집합이 사용되었다.

ABSTRACT

Multi-domain engineering systems include electrical, mechanical, hydraulic, pneumatic, and thermal components, making it difficult to design a system because of their complexity and inter domain nature. In order to obtain an optimal design, a unified design approach for each domain and an automated search method are required. This paper suggests a method for automatically synthesizing designs for multi-domain systems using the combination of bond graph that is domain independent and genetic programming that is well recognized as a powerful tool for open-ended search. To investigate the effect of proposed approach, an eigenvalue design problem is tested for some sample target sets of eigenvalues with different embryos.

키워드

멀티-도메인 시스템, Bond Graph, Genetic Programming, 자동 설계, 고유값

1. 서 론

멀티-도메인 동적 시스템은 전기, 기계, 전자장,

유체, 열 등과 같은 다수의 에너지 도메인을 포함하고 있다. 일례로 메카트로닉 시스템은 대표적인 멀티-도메인 시스템이며 전기, 전자와 기계분야가

*서경대학교 전자공학과

**안동대학교 전자정보산업학부

결합된 시스템이다.[1] 이와 같은 멀티-도메인 설계는 시스템이 복잡하고 해 공간이 구조적으로 제한되어 있지 않다는 특성과 단일 도메인 시스템에 대한 전통적인 설계 방법으로는 한계가 있는 문제점을 가지고 있다. 특히, 설계자의 경험에 많이 의존하는 디자인 특성상, 자동적인 설계 방법의 필요성이 높아지고 있는 시점에서, 새로운 접근방법이 요구되는 실정이다.[2]

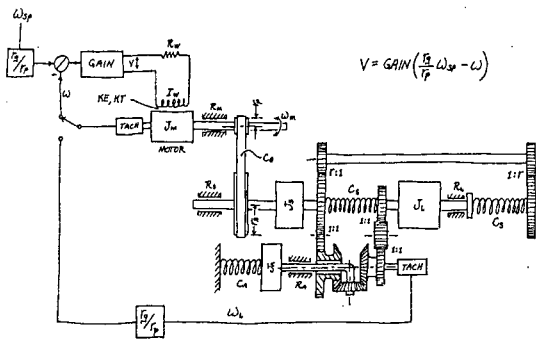


그림 1. 메카트로닉 시스템의 한 예-프린터 드라이버
Fig1. Example mechatronic system-the printer drive

본 논문에서는 이러한 멀티-도메인 시스템을 자동적으로 설계하기 위한 새로운 방법론을 제시한다. 이를 위하여 두 가지 핵심적인 접근법을 사용하였다. 첫째, 멀티 도메인 시스템을 일관된 도구로서 해석할 수 있는 본드 그래프(bond graph) [3,4]와, 둘째, 반복적이고 디자인 해의 공간이 매우 크고, 구조적으로 제한되지 않는(open-ended) 설계 특성을 만족시킬수 있는 진화연산의 일종인 Genetic Programming(유전 프로그래밍, GP)를 [5,6]을 결합하여 적용하였다. 즉, 동적 시스템을 탐색에 적합한 본드그래프로 모델링한 다음, 유전 프로그래밍(Genetic programming)을 이용하여 원하는 성능 조건이 만족될 때까지 이 모델을 진화시켜서 디자인 해를 얻는 것이다.

지금까지의 기존 접근법들은, Koza등[7]과 같이 단일-도메인 문제를 대상으로 하였거나, 멀티-도메인 환경에 적합한 본드 그래프(bond graph)를 사용하는 사례도 있으나 대부분, 모델링이나 해석의 일부 프로세스에만 사용되고 있고, 일부에서 디자인 해의 생성에 이를 이용하려는 시도가 있었으나, 모델 일부분의 변형에 그치는등 제한적이다.[8,9]

본 논문에서는 본드 그래프 모델의 성능해석을

위한 causality 검사루틴과 상태방정식의 자동적인 도출 방법을 구현하였다. 그리고 본드 그래프 생성을 위한 효율적인 GP 함수와 GP 탐색기법을 개발하였다. 또 자동설계시 시작점인 태아모델과 해의 구조와의 영향을 고찰하기 위해서 서로 다른 태아(embryo) 모델을 사용하였다. 제안된 설계방법의 효용성을 입증하기 위해서 고유값(eigenvalue) 설계 문제에 적용하였고, 2, 4, 6 개의 고유값을 가진 목표에 대해서 실험하였다.

II. 동적 시스템과 본드 그래프

2.1 동적 시스템

동적(dynamic) 시스템은 시간에 따라 시스템의 상태가 변하는 특성을 가지고 있으며, 다음 시점(t+1)에서의 시스템 상태가 전의 시점(t)에서의 상태에 영향을 받는다. 다음 식은 동적 시스템의 상태 방정식을 나타내며, X는 시스템의 상태를 나타내는 벡터이며, U는 입력 벡터, A, B는 각각 n*n 행렬이다.

$$X(t+1) = A \cdot X(t) + B \cdot U(t) \tag{1}$$

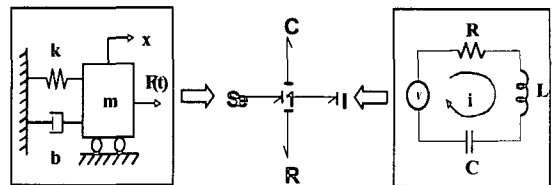


그림 2. 간단한 동적 시스템의 예와 본드그래프-기계적 시스템, 전기 회로

Fig. 2. Simple example of dynamic systems and their bond graph model-mechanical system, electrical circuit

동적시스템(dynamic system)의 예로는 각종 전기적인 회로와 기계적인 시스템, 그리고 유압, 열역학 시스템을 들수 있다. 다음 그림 1은 간단한 동적 시스템의 예이다. 단일 도메인 시스템의 예로써, 왼쪽은 스프링-댐퍼-질량으로 구성된 기계적 시스템이고, 오른쪽은 저항(R), 인덕터(L), 캐패시터(C)로 구성된 RLC 전기회로이다. 가운데 의 그림이 두 시스템의 등가적인 본드그래프 모델을 나타낸다. 그림 2는 다중 도메인의 예로써, 메카트로닉 시스템의 일종인 프린터 드라이버를 나타낸다.

그림 3은 그림 2에 대한 본드 그래프 모델이다.

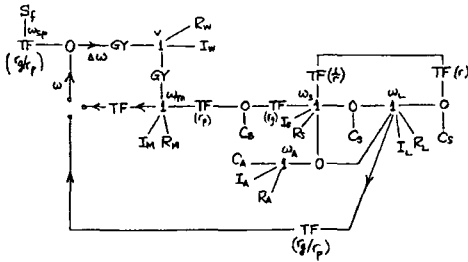


그림 3. 프린터 드라이브의 본드그래프 모델
Fig. 3. Bond graph model for printer drive

2.2 본드 그래프

본드 그래프(bond graph)는 모델링 도구로서 동적 시스템의 모델링과 해석에 대한 일관된 접근법을 제공한다. 본드 그래프는 모델링이 쉽고, 시스템의 특성을 명확히 나타낼수 있으며, 동적 시스템의 상태방정식을 쉽게 유도할수 있는 장점이 있다. 또한 컴퓨터를 이용하기 적합한 형태를 가지고 있다.[2]

동적인 시스템에서 일률(power) $P(t)$ 는 작력(effort)과 흐름(flow) 변수의 곱으로 정의된다.

$$P(t) = e(t)f(t) \tag{2}$$

참고로 시스템에 따른 작력과 흐름 변수의 종류는 표 1과 같다. 모멘텀 $p(t)$ 과 변위 $q(t)$ 두가지 종류의 변수가 동적시스템을 기술하는데 사용된다.

모멘텀 $p(t)$ 는 다음 식 (3) 과 같이 정의된다.

$$p(t) = \int^t e(t)dt = p_0 + \int_{t_0}^t e(t)dt \tag{3}$$

같은 방법으로 변위 $q(t)$ 는 다음과 같이 정의된다.

$$q(t) = \int^t f(t)dt = q_0 + \int_{t_0}^t f(t)dt \tag{4}$$

따라서 에너지 $E(t)$ 는 식 (5) 로 표시될 수 있다.

$$E(t) = \int^t P(t)dt = \int^t e(t)f(t)dt \tag{5}$$

표1. 시스템 종류에 따른 effort 와 flow 변수들
Table 1. Effort and flow variables for different systems

Domain	Effort, $e(t)$	Flow, $f(t)$
기계적 병진	힘, $F(t)$	속도, $V(t)$
기계적 회전	토크, $\tau(t)$	각속도, $\omega(t)$
유압	압력, $P(t)$	체적유량, $Q(t)$
전기	전압, $e(t)$	전류, $i(t)$

본드 그래프는 본드(bond) 와 노드의 상호 연결된 조합으로 구성된다. 노드는 다음과 같이 몇가지 요소로 나누어진다. 첫째 에너지 발생요소로서 Se 와 Sf가 있다. Se는 작력 소스(effort source) 로서 작력(effort)을 입력으로 가지며, 전압소스, 힘, 압력 소스 등을 예로 들 수 있다. Sf 는 흐름 소스(flow source) 로서 흐름을 입력으로 가지며, 전류 소스, 속도, 유속을 예로 들 수 있다. 둘째, 에너지 저장 및 발산 요소로서, C, I, R이 있다. C는 캐퍼시턴스 요소로서 작력과 변위(displacement)간의 정적인 관계가 존재한다. 물리적인 스프링, 전기적인 캐퍼시턴스가 이에 해당한다. I는 인덕턴스 흐름과 모멘텀과의 정적 관계를 포함하며, 물리적인 관성과 전기적인 인덕턴스를 예로 들 수 있다. R은 저항 요소로서 작력과 흐름간의 정적인 관계가 존재한다. 물리적인 댐퍼, 전기적인 저항이 이에 해당한다. 그리고, 접합요소로서 0-접합(junction-0) 과 1-접합(junction-1) 이 존재한다. 0-접합에서는 작력변수가 연결된 모든 본드에서 공통이다. 1-접합에서는 흐름 변수가 연결된 모든 본드에서 공통인 성질을 가지고 있다. 이 밖에 에너지 변환요소로서 트랜스포머 TF 와 자이레이터 GY 가 있다.

본드상의 막대 바는 causal stroke 라고 하며 작력(effort) 과 흐름(flow)이 본드 상에서 어떻게 동시에 결정되는가를 나타낸다.(그림 4) 원인-인과 관계는 작력(effort) 과 흐름(flow)에 대한 서로 반대 방향으로 표시된다. 이 성질은 초기 단계에서 매우 간단히 디자인 해의 적합성을 결정하는 중요한 역할을 한다.

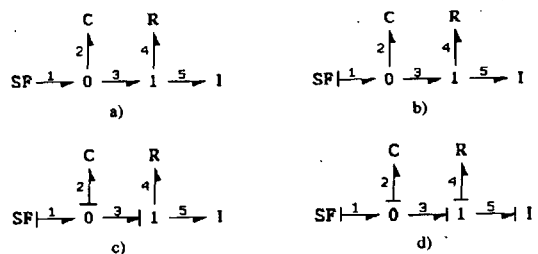


그림 4. Causality 지정 절차
Fig. 4. Assignment procedure of Causality



그림 5. causal stroke 의 의미
Fig. 5. The meaning of causal stroke

본드 그래프로부터 상태 방정식을 유도하는 과정은 다음의 두 단계로 나뉜다 - 1) 인과관계 분석 (causality analysis), 2) 상태방정식 구성(state equation formulation). 먼저 인과관계 분석 (causality analysis) 에서는 요소간의 인과(causal) 관계와 힘의 흐름을 중심으로 모델의 인과관계성 (causality) 위반 여부를 간단하게 판별할 수 있으며, 여기서 만족된 모델만 두 번째 단계인 상태방정식 구성(state equation formulation) 을 진행할 수 있다. 따라서, 동적인 시뮬레이션이 가능하지 않은 모델은 단계 1에서 제거됨으로 전체적인 시뮬레이션 시간을 상당히 단축할 수 있다. 인과 관계 분석 절차는 다음과 같다.

- [step 1] 임의의 S_e 와 S_f 를 선택하고, 필요한 인과관계성을 설정한다. 이어서 모든 0, 1, TF, GY의 제한 규정을 이용하여 인과관계성을 확장한다. (그림 5b)
- [step 2] step 1을 모든 에너지 소스에 대해서 실행한다.
- [step 3] 임의의 C 나 I를 선택하고 정해진 인과관계성을 설정한다. 다시 이를 모든 0, 1, TF, GY의 제한 규정을 이용하여 인과관계성을 확장한다.(그림 5c)
- [step 4] step 3을 모든 C 와 I 요소에 대해서 반복한다.
- [step 5] 아직 설정되지 않은 임의의 R 요소를 선택해서 임의의 인과관계성을 설정한다. 다시 이를 모든 0, 1, TF, GY의 제한 규정을 이용하여 인과관계성을 확장한다. (그림 5d)

step 4. step 5을 모든 R 요소에 대해서 반복한다. 단계 2에서는 causality analysis 와 0,1-접합에서의 에너지 흐름의 합으로부터 얻어진 관계 행렬을 이용하여 상태방정식을 자동적으로 얻을 수 있다. (자세한 내용은 [3] 참고).

III. 본드그래프의 진화

3.1 GA/GP

유전 알고리즘(Genetic algorithm, GA)은 탐색의 강인성으로 인해 최적화 문제에 많이 응용되고 있다. Genetic programming(GP) 은[5,6] 큰 부류에서 GA 와 함께 진화 연산(evolutionary computation)에 속하지만, GA와는 몇가지 다른 특징이 있다. 가장 큰 다른점은 GA가 개체를 표현할 때 스트링을 사용하는데 비해 GP는 트리를 사용한다. 이때 트리에서의 각 노드가 하나의 함수를 나타내며, 각 개체는 함수의 집합, 즉 컴퓨터 프로그램으로 구성된다. 다시 말해 GP 는 각 개체가 직접 해를 표현하는 것이 아니라, 해를 생성해 주는 하나의 프로그램을 표현한다. 또 다른 차이점은 대부분의 GA가 고정크기의 염색체를 사용하는데 비해 GP는 가변크기의 염색체를 사용한다.

3.2 본드 그래프와 GP 의 결합

본드 그래프를 디자인에 적용시의 장점은 다음 두 가지로 볼 수 있다. 첫째는 디자인 해의 평가를 효율적으로 할 수 있다는 점이고, 둘째는 본드 그래프가 가지고 있는 자체적인 특성인 본드와 노드의 조합으로 새로운 디자인 해를 생성하는데 적합하다는 점이다. 두 번째의 특성을 부연 설명하면, 시스템의 상태식에 무관하게 본드 그래프 모델의 구조적인 변경이 가능하며, 이를 통해 임의의 시스템 모델이 본드와 노드의 조합으로 자유롭게 생성될 수 있다는 점이다.

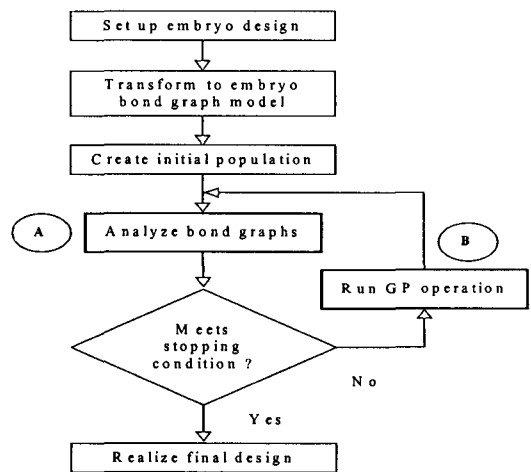


그림 6. (a) Bond/GP 디자인 알고리즘
Fig 6. (a) The algorithm of Bond/GP design

그러나 수많은 디자인 해를 반복적으로 생성하고 평가해야 하는 공학적 설계의 특성상, 대규모의 탐색이 가능한 자동 설계 방법이 필요하다. 따라서 디자인 해의 생성에 대한 크기적 제한이 없는 (open-ended) GP를 본드 그래프에 결합시켜 대규모의 공간을 탐색하면서 디자인 해를 최적화 할수 있는 기법을 제공할 수 있다.

본드 그래프(bond graph)와 GP를 이용하여 멀티 도메인 동적시스템에 대한 디자인 해를 자동적으로 생성해주는 전체 알고리즘이 그림 6(a)와 6(b)에 나와 있다. 디자인 시 하나의 태아(embryo) 모델에서 출발하여 이 모델을 진화시켜(evolve) 하나의 완성된 디자인 해를 구하는 방법이다. 먼저 태아(embryo) 디자인을 정하고, 이를 본드 그래프 모델로 변환한 다음, 이를 토대로 하여 초기 본드 그래프 모델 개체들을 생성한다. 다음 각 개체 모델들을 해석하여 평가하고(자세한 설명은 그림 6(b)의 윗부분), 조건이 만족될때까지 GP 오퍼레이션을(자세한 설명은 그림 6(b)의 아래부분) 수행한다.

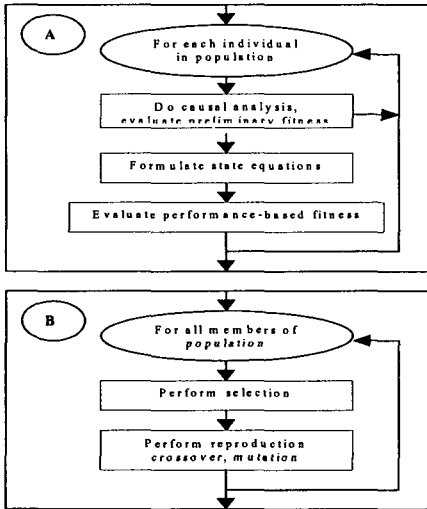


그림 6. (b) Bond/GP 디자인 알고리즘
Fig 6. (b) The algorithm of Bond/GP design

IV. GP 에 의한 본드그래프 모델의 생성

본드 그래프 모델은 생성된 GP 트리에 따라 깊이 우선 탐색으로 GP 함수와 터미널을 특정 본드

와 노드에 실행시킴으로서 성장될수(grow) 있다. 여기에서 사용된 GP 함수와 터미널에 대한 종류는 다음과 같다.(표 2)

구체적인 GP 함수의 실행 과정은 다음과 같다. 먼저 본드에 작용하는 함수의 한 예로서 insert_J0에 대한 설명이 그림 7에 나와있다. 이 함수는 작용하는 0-접합(0-junction)을 수정가능한 해당 본드에 삽입한다. 그림 7의 왼쪽 그림에서 노드 1과 R을 연결하는 본드에 0-접합을 삽입하면, 결과로 새로운 접합-0 과 본드가 추가된다.(그림 7의 오른쪽) 부수적으로 새로이 GP 함수가 작용될수 있는 3개의 수정가능한 지점(modifiable site)가 존재하게 된다. 그림 7의 아래는 이 함수의 트리 표현이며, 이때 트리의 노드는 GP 함수를 의미한다. 각 가지의 숫자가 수정가능한 지점에 대응된다. 그리고 각 가지에 연결되는 다음 트리의 노드에 올 수 있는 GP 함수의 형태가 제한적으로 결정된다.

표 2. GP 터미널과 함수
Table 2. GP terminals and functions

Name	#Args	Description
add_C	4	Add a C element to a junction
add_I	4	Add an I element to a junction
add_R	4	Add an R element to a junction
insert_J0	3	Insert a 0-junction in a bond
insert_J1	3	Insert a 1-junction in a bond
replace_C	2	Replace the current element with a C element
replace_I	2	Replace the current element with an I element
replace_R	2	Replace the current element with an R element
+	2	Add two ERCs
-	2	Subtract two ERCs
enda	0	End terminal for add element
endi	0	End terminal for insert junction
endr	0	End terminal for replace element
erc	0	Ephemeral random constant (ERC)

그림 8에는 기존의 1-접합에 R 노드를 추가하는 add_R 함수에 대한 설명이 나와있다. 이 함수는 주어진 1-접합에 R 요소를 추가한다. R 요소의 파라미터 값은 ERC(ephemeral random constant)의 생성을 통해서 직접 값이 할당되거나, 몇 개의 값이 +/- 연산을 통해 얻어지기도 한다.

그림 9에는 실제 생성된 GP 트리의 예가 나와 있다. 이 GP 트리를 태아모델에 적용시켜 그림 10의 진화된 본드그래프를 얻는다. 그림 10에서 점선

밖의 부분은 원래 주어진 태아모델중 고정된 부분이고, 점선내에서 가장 아래의 3개의 수정가능한 지점 - 1-접합(11) 과 0접합(01) 및 그 사이의 본드(a) - 은 태아모델의 일부분이나 진화될수 있는 변형가능한 부분이다.

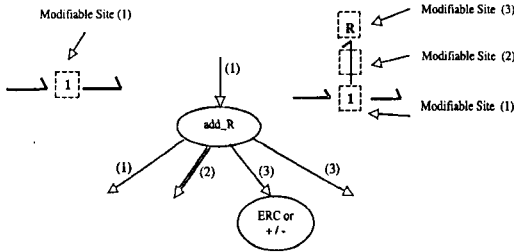


그림 7 add_R function
Fig. 7 The add_R function

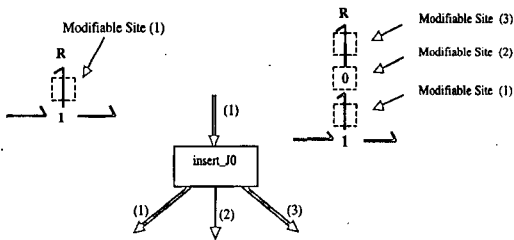


그림 8 insert_J0 function
그림 8 The insert_J0 function

태아모델의(그림 10) 3개 수정가능지점(modifiable sites)에 대응하여 루트 노드의(그림 9) 가지수와 종류가 결정된다. 1 과 2 로 표기된 단일선의 가지는 본드그래프의 노드에 작용되는 것을 의미하고, a 로 표시된 이중선의 가지는 본드그래프의 본드(아크에 해당)에 작용되는 것을 의미한다. 가지 1을 따라가면 `add_I` 의해서 I 요소가 1-접합(그림 10의 11)에 추가된다. 루트 노드에서 가운데의 가지 a 에는 바로 `end` 노드가 오면서 더 이상 확장이 되지 않는다. 오른쪽의 가지 2에는 `add_C` 노드가 오면서, 그림과 같이 그 아래로 계속적인 가지의 확장이 수행된다.

그림 9의 GP 트리를 깊이우선탐색으로 각 노드를 수행하면, 그림 10과 같은 본드그래프를 얻을수 있다.

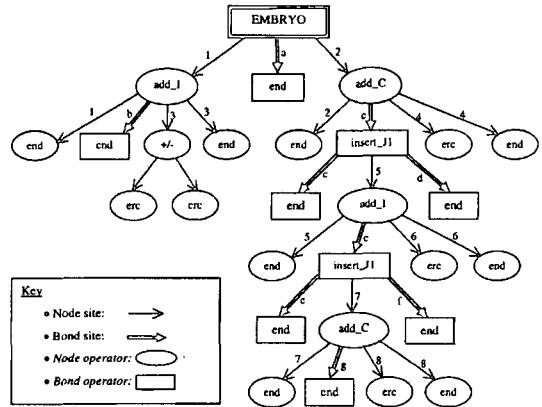


그림 9. GP 트리의 예
Fig 9. Example of GP tree

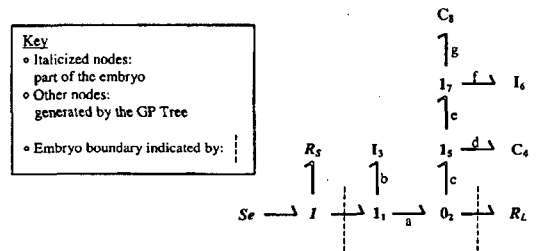


그림 10. 그림 9의 GP 트리에 의한 생성된 본드그래프

Fig 10. Bond Graph generated by the example of GP tree of Fig. 9

4.2 태아 모델의 선택

자동적인 설계를 위하여 태아(embryo) 모델은 가장 간단한 구성을 가진다. 그러나 본드 그래프의 특성상 수정지점이 노드(C, I, R 과 같은 요소들)와 본드(아크)에 따라 수행되는 동작이 다르므로 선택의 주의가 요구된다. 태아 본드 그래프 모델은 그림 11과 같이 두 가지가 사용되었다. SE 와 R1, R2 는 기본적인 입출력 구성을 위한 최소 구성요소이므로 당연히 포함되어야 한다.

그림 11 a) 는 하나의 modifiable site를, 그림 11 b) 는 세개의 modifiable site를 포함한다. Modifiable site 는 점선으로 된 박스로 되어있으며, 초기의 수정가능 지점을 나타낸다. 즉, 이 지점에서 C,I,R 같은 요소의 부착이나, 0,1 junction 의 삽입이 이루어지며, 주어진 GP 트리의 해석이 끝날때까지 계속적으로 GP 트리에서 명시된 구축 절차에 따라 부착과 삽입의 오퍼레이션이 수행된

다. Modifiable site 는 주어진 부착과 삽입 오퍼레이션의 규칙에 따라 오퍼레이션 수행후 계속적으로 증가될수 있다. 이를 기반으로 임의의 구조를 가진 본드그래프 모델이 생성될수 있다.

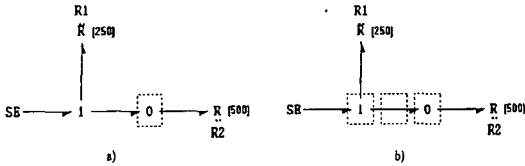


그림 11. 두 종류의 태아 본드그래프 모델, a) 한 개의 modifiable site, b) 두개의 modifiable site
Fig. 11. Two types of embryo bond graph model with a) One modifiable site, b) Three modifiable sites

수정가능지점이 노드일 경우는 노드에 관한 오퍼레이션, 즉 add_C, add_I, add_R 등만이 수행될 수 있다. 노드 오퍼레이션 수행후 결과로서 두 개의 수정가능한 노드와 하나의 수정가능한 본드가 생성된다. 수정가능지점이 본드일 경우는 본드에 관한 오퍼레이션, 즉 ins_J0, ins_J1 등만이 수행될 수 있다. 본드 오퍼레이션 수행후 결과로서는 한 개의 수정가능한 노드와 두개의 수정가능한 본드가 생성된다.

V. Case study—Eigenvalue 설계 문제

5.1 문제 정의 및 실험

고유값(Eigenvalue) 할당문제는 제어시스템 설계분야에서 큰 관심을 끌어왔다. 시스템 설계에서 안정성을 확보하는 것이 매우 중요하고 실용적인 문제이다. 아래 예제에서 일련의 목표 고유값이 주어지고 이들 고유값을 가지는 본드 그래프 모델이 생성된다. 다음과 같은 3개의 집합으로 된 목표 고유값에 대해 lil-gp[10]를 사용하여 Pentium IV 2.8GHz PC 상에서 수행을 시켰으며, 태아의 각기 다른 선택에 따른 영향을 보기위해 앞 절에서 언급한 두 종류의 태아 모델이 새로이 실험되었다.

- 1) $-1 \pm 2j$
- 2) $-1 \pm 2j, -2 \pm j$
- 3) $-1 \pm 2j, -2 \pm j, 3 \pm 0.5j$

그리고 사용된 GP 파라미터는 다음과 같다.

- 세대수: 100 - 500
개체수: 100 - 500

- 초기개체생성: half_and_half
최대깊이(Max depth): 17
초기깊이(Initial depth): 2-6
선택: Tournament (size=7)
교배(Crossover): 0.9
돌연변이(Mutation): 0.1

5.2 결과 및 분석

각 목표값에 대해서 30번 이상의 반복 실행이 수행되었고, 그중 일부 결과들이 그림 12-17에 설명되어 있다. 그림 12는 목표값 $-1 \pm 2j$ 에 대해서 얻어진 고유값 해와 목표값에 대한 오차가 구해진 본드그래프와 함께 나와 있다. 평균거리오차는 (average distance error)는 총거리오차를 구한다면 이를 고유값 수로 나눈 것이다. 두 개의 복소 고유값에 대해서 구해진 결과가 매우 오차가 적음을 알 수 있다. 이어서 태아 모델로부터 추가된 junction, element, bond 의 수와 선택된 C, I, R 요소에 대한 파라미터 값이 나와 있다.

오른쪽에는 진화 연산에 의해 얻어진 본드 그래프 모델의 구조가 나와 있다. 태아 본드 그래프에서 점선 박스로 되어있는 0 junction 이 수정가능 지점으로 주어진 상태에서, 여기에 GP 의 수행 결과 각각 하나씩의 C와 I, 그리고 R 요소가 결합되어 주어진 고유값 해를 가지는 모델이 생성되었음을 알 수 있다. 고유값의 차수를 결정하는 C와 I가 각각 하나씩임을 확인할 수 있다.

그림 13에는 같은 목표값 $-1 \pm 2j$ 에 대해서 수정가능지점이 3개인 다른 태아모델을 사용해서 얻어진 결과가 나와 있다. 비록 3개의 수정가능지점이 주어졌지만, 두개의 상태 벡터만 필요하므로, 한 지점에서는 비슷한 성능을 가진 디자인들 중에서 구조의 구현성과 비용, 그리고 선호도등을 고려하여 선택할수 있으면 바람직하다고 할수 있다. 차수가 낮은 경우에는 구조상의 어떤 제한에 의한 차이가 나기 어려우나, 다음에 나오는 좀 더 높은 차수에서는 다른 태아모델에 의한 영향 차이를 볼수 있다.

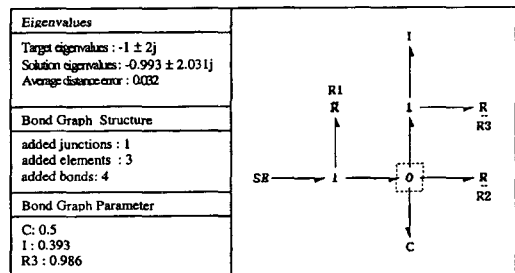


그림 12. 두 개의 고유값 결과-하나의 modifiable site
Fig. 12. Two-eigenvalues result with one modifiable site

그림 14는 4개의 목표값 $-1 \pm 2j$, $-2 \pm j$ 에 대해서 얻어진 고유값 해를 보여준다. 2개의 목표값에 비해 구조가 좀 더 복잡해졌음을 알 수 있고, 상태벡터인 C 와 I 가 각각 2개씩 새로이 추가되어서 고유값의 수가 4개와 일치함을 보여준다. 태아 모델에 새로 추가된 형태가 C, I, R 이 같은 junction 0 또는 1에 연결되어 하나의 패턴을 이루고 있음이 흥미롭다. 이렇게 구해진 본드 그래프 모델은 각 도메인의 실제적인 구성요소로 직접 변환되어 최종적인 설계 해가 얻어질 수 있다.

그림 15에는 3개의 수정가능지점을 가진 태아 모델에 대한 4개의 고유값 결과가 나와 있다. 그림 14의 1개의 수정가능지점에 비해 균형 잡힌 구조를 가지고 있다. 즉 왼쪽의 수정가능한 junction 1 에 C와 I 요소가 추가 되었다고, 오른쪽의 junction 0 에 C, I, R 요소가 추가 되었다. 고유값 설계 문제가 근사화된 최적해를 구하는 문제이므로 요구되는 오차 범위내에서는 해의 존재 공간에 따라 몇가지의 해가 나올 수 있다. 수정가능지점이 하나인 태아 모델보다는 수정가능지점이 세개인 태아 모델이 좀 더 폭 넓은 해의 탐색이 가능해 보인다.

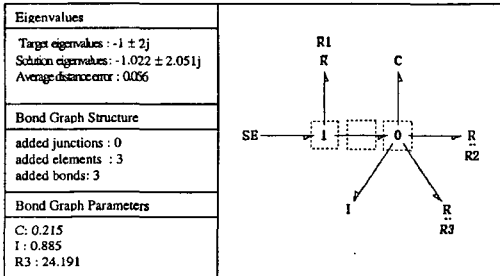


그림 13. 두 개의 고유값 결과-세개의 modifiable site
Fig. 13. Two-eigenvalues result with three modifiable sites

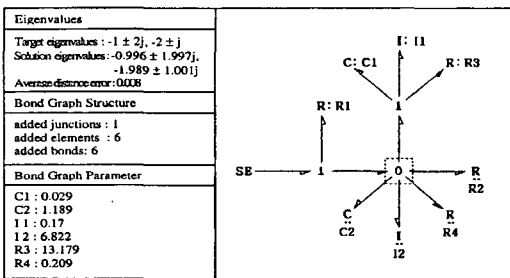


그림 14 4 개의 고유값 결과-하나의 modifiable site
Fig. 14. Four-eigenvalue result from one modifiable site

그림 16 와 17에는 목표값 $-1 \pm 2j, -2 \pm j, 3 \pm 0.5j$ 인 6개 고유값 설계에 대한 결과가 두개의 태아 모델에 대해 나와 있다. 고유값의 차수가 높아짐에 따라 문제의 난이도가 매우 높아졌고, 해의 공간도 더욱 복잡해 졌다. 2개와 4개 고유값 문제와는 달리 구해진 결과의 구조들이 좀 더 다양해 졌고, 그림 16과 17는 그 중에서 한 예들을 나타낸 것이다.

그림 16의 경우는 한지점에서 고유값의 차수에 관계된 6개의 상태벡터와 고유값에 관계된 R 요소들이 0-접합 통해 연결되므로 구조의 확장이 제한적일 수 있다. 즉 수정가능지점이 하나인 경우는 어떤 특정 구조를 가진 본드 그래프 모델을 설계해야 될 경우 원천적으로 그 구조에 도달할 수 없는 경우가 생길 수 있다. 그림 17와 같이 수정가능지점이 3개인 경우는 어느정도 이를 피할 수 있다.

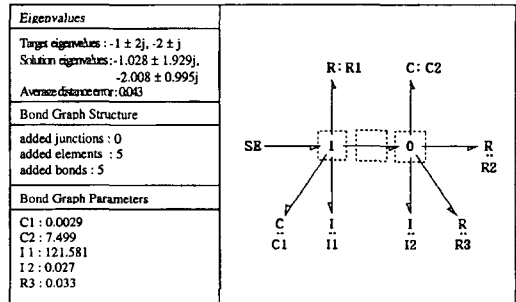


그림 15. 4 개의 고유값 결과-세개의 modifiable site
Fig. 15. Four-eigenvalue result from three modifiable site

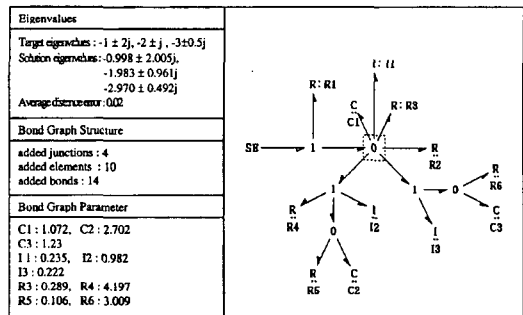


그림 16. 6 개의 고유값 결과-하나의 modifiable site
Fig. 16. Six-eigenvalue result with one modifiable site

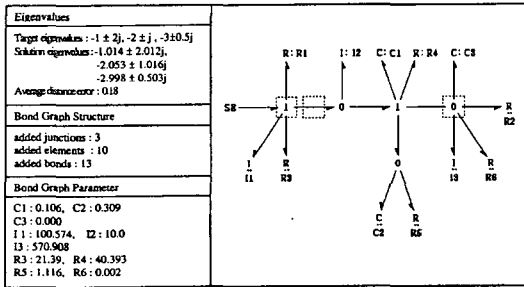


그림 17. 6 개의 고유값 결과-세개의 modifiable site
Fig 17. Six-eigenvalue result with three modifiable sites

VI. 결 론

본 연구에서는 본드그래프와 Genetic Programming을 이용한 멀티-도메인 시스템을 자동적으로 설계하는 새로운 접근법이 제안되었고, 여러가지 고유값 설계문제에 대하여 서로 다른 태아 모델별로 적용하여 효용성을 보였다. 비록 고유값의 차수가 최대 6 차까지만 실험되었지만, 얻어진 고유값 결과들이 매우 정확함을 알 수 있다. 또한 사용된 태아모델과 생성된 구조의 형태와의 연관성을 고찰할 수 있었다. 또한 탐색의 효율성이 개선된다면 좀 더 높은 차수의 문제에도 좋은 결과를 얻을 수 있으리라 생각된다.

앞으로 실제적인 공학 시스템 설계에 적용이 되어야 하고, 본드 그래프 모델에서 해당 도메인에 대한 실제 디자인을 도출해야(physical realization) 하는 문제가 남아 있지만, 멀티-도메인의 동적 시스템을 자동적으로 설계하는 방법론의 제시와 고유값 설계 문제의 적용은 진화 연산을 이용한 공학 설계를 위한 의미 있는 시도라고 볼 수 있다.

참고문헌

[1] K. Youcef-Toumi, Modeling, Design, and Control Integration: A necessary Step in Mechatronics, IEEE/ASME Trans. Mechatronics, vol. 1, no.1, 1996, pp. 29-38

[2] E. Coelingh, T. J. A. de Vries, J. V. Amerongen, "Automated Performance Assessment of Mechatronic Motion Systems During the Conceptual Design Stage,"

Proceedings of the 3rd International Conference on Advanced Mechatronics, Okayama, Japan, pp.472-477, 1998

[3] D. C. Karnopp, R. C. Rosenberg, D. L. Margolis, [1999] System Dynamics, A Unified Approach, 3rd Ed., John Wiley & Sons

[4] R.C.Rosenberg, "Reflections on Engineering Systems and Bond Graphs," Trans. ASME J. Dynamic Systems, Measurements and Control, V.115, p.242-251. 1993b

[5] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992

[6] J. R. Koza, Genetic Programming II, Automatic Discovery of Reusable Programs, MIT Press, 1994.

[7] J. R. Koza, F. H. Bennet, D. Andre, M. A. Keane, F. Dunlap, Automate Synthesis of Analog Electrical Circuits by Means of Genetic Programming, IEEE Trans. on Evolutionary Computation, vol. 1, no. 2, pp.109-128. 1997

[8] J. E. E. Sharpe, R. H. Bracewell, "The Use of Bond Graph Reasoning for the Design of Interdisciplinary Schemes", 1995 International Conference on Bond Graph Modeling and Simulation, pp.116-121

[9] E. Tay, W. Flowers and J. Barrus, "Automated Generation and Analysis of Dynamic System Designs", Research in Engineering Design, vol 10, 1998, pp. 15-29.

[10] D. Zongker, W. Punch, lil-gp 1.1 Users Manual, Michigan State University, 1996

저자소개

서기성(Kisung Seo)



1993년 연세대학교 대학원 전
기공학과 공학박사
1993년~1998년 서경대학교 전
자공학과 조교수
1999년~현재 Michigan State
University, Case Center of
Computer Aided Engineering and
Manufacturing, Research Associate

※ 관심분야 : 유전 알고리즘, 유전 프로그래밍,
Evolutionary design

박세현(Se-Hyun Park)

1980년 경북대학교 전자공학과 학사

1982년 경북대학교 전자공학과 석사

1985년 아주대학교 전자공학과 박사

1997년 국민 포장 수여

현재 안동대학교 전자정보산업학부 교수

1997년~1999년 국립 안동대학교 공과대학 학장

1999년~2000년 미시간 주립대학 전기컴퓨터공학과 겸임 교수

※관심분야 : FPGA 설계, 컴퓨터구조, 하드웨어 유전자 알고리즘