工學碩士 學位論文

SA와 RCGA를 이용한 차륜형 도립진자 이동로봇의 모델링

Modeling of a Wheeled Inverted Pendulum Type Mobile Robot Using a SA and RCGA

指導敎授 河 潤 秀

2001年 2月

韓國海洋大學校 大學院

制御計測工學科

白 雲 鶴

목 차

Abstract iii

제 1 장 서 론 1

제 2 장 대상시스템의 모델링 4

2.1 **차륜형 도립진자 이동로봇의 구조** 4

2.2 **차륜형 도립진자 이동로봇의**
제어에 있어서의 난점 6

2.3 **비선형 모델** 7

제 3 장 SA를 접목한 RCGA기반 모델선형화 13

3.1 RCGA 13

3.1.1 **염색체 표현** 16

3.1.2 **초기 집단의 생성** 16

3.1.3 **재생산 연산자** 17

3.1.4 **교배** 17

3.1.5 **돌연변이** 18

3.1.6 **적합도 평가** 20

3.2 SA 21

3.2.1 **초기온도의 설정** 25

3.2.2 **온도감소 계수의 설정** 25

3.2.3 **종료조건** 26
3.3 SA를 **접목한** RCGA 26
3.4 **선형모델의 최적 파라미터 추정** 28

제 4 장 검증시뮬레이션 **36**
4.1 **시뮬레이션 조건** 36
4.2 **결과 및 검토** 36

제 5 장 결론 **41**

참고문헌 **42**

Modeling of a Wheeled Inverted Pendulum Type Mobile Robot Using a SA and RCGA

Woon-Hak Baek

Department of Control & Instrumentation Engineering, Graduate school, Korea Maritime University

ABSTRACT

This paper describes the modeling of a wheeled inverted pendulum type mobile robot driven by two different wheels for the posture and velocity control. In order to control its posture and velocity, the optimal linearized modeling is quite crucial.

Meta-heuristics is a term used to characterize a number of methods which have been proven to be practical and effective algorithms for solving nonlinear problems.

This paper adapted the wheeled inverted pendulum type mobile robot which is typically nonlinear systems identification and linearization techniques, using a real-coded genetic algorithm. The algorithm is finely tuned by simulated annealing, which yields a faster convergence and a more accurate search.

By applying this method, the nonlinear model is transformed into a completely linearized system.

제 1장 서 론

최근 정보산업사회의 도래와 더불어 로봇에 대한 관심과 중요성이 날로 높아져 가고 있다. 그 중에서도 이동로봇에 관한 수많은 연구와 개발이 이루어지고 있으며 그 형태도 점차 다양해 지고있다. 이러한 연구결과로부터 파생된 것 중 하나가 도립진자형 이동로봇이다.

도립진자는 1965년 기계적인 불안정한 모드를 안정한 모드로 유지시키기 위해서 Kapitza에 의해서 처음으로 설계 제작되어 졌다[1]. 그후 도립진자는 새로운 제어이론의 우수성을 입증하는데 그 대상으로 자주 이용되어져 왔다[2][3][4]. 그 결과 다양한 형태의 도립진자의 등장과 안정화기술이 확립되었다.

이러한 조류에 부응하여 연구 개발된 것이 차륜형 도립진자 이동로봇이다. 차륜형 도립진자이동로봇은 삼·사륜형 또는 그 이상의 차륜을 가진 이동로봇에 있어서 자세보조륜(caster)에 의한 문제 즉, 노면의 상태에 따라 자세보조륜으로부터 큰 불규칙적 외란을 받을 수 있고, 또한 경사진 도로를 주행할 경우에 무게중심이 뒤쪽으로 치우치는 문제점을 해결할 수 있는 이동체로서 연구 개발되어지고 있다[5][6]. 또한 일륜과 보행로봇에 비해 제어가 쉬우며, 좁은 장소에서도 회전이 가능하다는 장점을 가지고 있다.

그러나, 이러한 형태의 로봇을 제어하는 데는 몇 가지 제약이 따른다. 차륜형 도립진자 이동로봇은 자기 자신의 균형을 스스로 유지하면서 요구되는 속도로 이동해야 한다. 따라서 차륜형 도립진자 이동로봇은 요구되는 자세와 속도를 동시에 만족시킬 수 있는 제어시스템 설계가 필수적이지만, 단입력 다출력 시스템이므로 자세 및 속도

모두 정상편차없이 제어하는 것은 불가능하다. 또한 이들의 제어를 위해 학습에 의한 제어, 퍼지제어에 등과 같은 비선형 제어기법이 고려되어질 수 있으나, 자립시스템의 경우 공급전원과 중량등을 고려해 마이크로프로세서등이 탑재되므로 계산능력에 있어서 많은 제약을 동반할 수가 있어 부담이 될 수가 있다. 게다가 이러한 형태의 로봇의 자세검출에 흔히 이용되고 있는 자이로 센서는 포텐쇼 미터, 엔코드, 가속도 센서등의 자세검출센서에 비해 응답속도가 뛰어나고 정도가 높으나 온도 드리프트를 가지고 있기 때문에 이로인한 오차를 보상할 필요가 있다. 이러한 이유에서 차륜형 도립진자 이동로봇의 제어기는 구성이 간단하면서도 성능이 우수한 것이 요구되어진다. 따라서 차륜형 도립진자 이동로봇의 동특성과 유사한 특성을 가지는 선형모델의 도출은 의미있는 일이라 할 수 있다. 이와 관련하여 기존의 모델링 방법 즉, 직립자세 근방에서 테일러급수 전개함으로써 얻은 모델에 근거해 설계된 제어기는 직립자세 근방에서의 자세제어성능은 우수하나 고속주행시에는 자세가 직립상태에서 멀어짐으로 문제가 된다. 이러한 문제를 보완하기 위해서는 직립상태에서부터 고속주행시까지의 넓은 운전 범위에 있어서 실제 시스템의 동특성과 유사한 선형모델의 도출이 요구되어 진다.

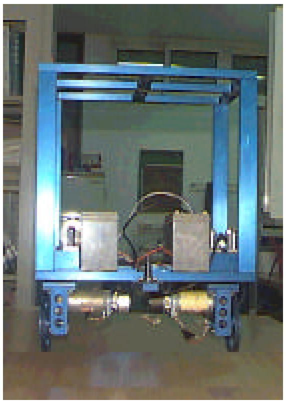
따라서 본 논문에서는 준 최적해 또는 최적해로의 이동이 용이한 RCGA와 등반능력이 우수한 SA를 접목함으로써 차륜형 도립진자 이동로봇의 자세 및 속도제어를 위한 선형모델의 최적파라미터를 추정하는 알고리즘을 제안한다. 또한 제안된 방법에 의해 얻어진 선형모델과 기존의 테일러 급수전개에 의한 선형화 모델, RCGA에 의해 얻어진 선형모델에 동일한 입력을 인가하고 이들의 응답을 실제 시스템의 응답과 비교함으로서 그 유효성을 입증한다.

본 논문의 구성으로는 다음과 같다. 제 2 장에서는 대상시스템의 모델링에 대해서 기술하고, 제3장에서는 SA와 RCGA를 이용한 모델 선형화에 대해서 기술한다. 제 4장에서는 검증시뮬레이션을 통해 제안된 방법의 타당성을 검토하고, 제 5장에서 결론을 맺는다.

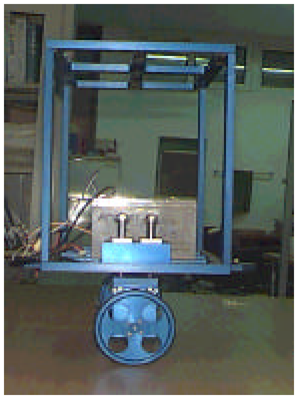
제 2장 대상 시스템의 모델링

2.1 차륜형 도립진자 이동로봇의 구조

사진 2.1은 본 연구에서 취급하게 될 제어대상인 차륜형 도립진자 이동로봇을 나타낸다. (a)는 차륜형 도립진자 이동로봇을 정면에서 본 것이고, (b)는 측면에서 본 것이다.



(a)



(b)

사진 2.1 차륜형 도립진자 이동로봇

Photo. 2.1 Wheeled inverted pendulum type mobile robot

사진에서 알 수 있듯이 대상시스템은 전후에 자세보조를 위한 보조륜이 장착되어져 있지 않으며, 본체 중심의 좌우에 있는 차륜만으로 스스로 균형을 유지하게 된다. 좌우 차륜은 감속 기어를 거쳐 독립된 DC모터에 의해 구동되며, 모터회전축에 회전수를 계측할 수 있는 엔코더가 장착되어져 있다.

이와 같은 차륜형 도립진자 이동로봇의 모델은 로봇이 균형을 유지하면서 평면상을 직선 주행할 때, 축상에 대하여 회전운동을 하는 1차원의 도립진자로 볼 수가 있다. 하지만 차륜형 도립진자 이동로봇이 2차원 평면상에서 빠른 속도로 이동하면서 방향을 전환할 때에는 엄밀하게 1차원 도립진자 모델로서 취급 할 수가 없다. 그러나 로봇의 이동속도를 충분히 감속한 후 회전한다면 본체의 자세각이 그다지 크게 되지 않으므로 본체의 회전각속도에 의한 원심력등이 자세에 미치는 영향은 적다. 따라서 본 논문에서는 차륜형 도립진자 이동로봇을 1차원 도립진자 모델로 취급한다.

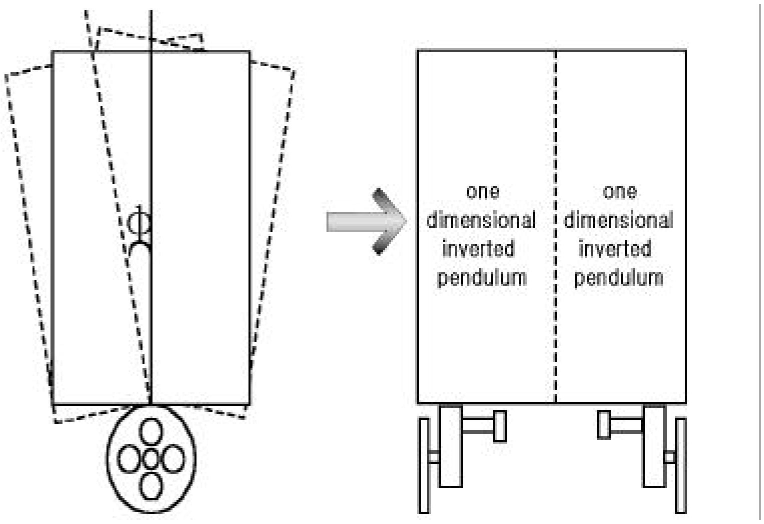


그림 2.1 차륜형 도립진자 이동로봇의 방향전환 동작

Fig. 2.1 Wheeled inverted pendulum type mobile robot in turning motion

2.2 차륜형 도립진자 이동로봇의 제어에 있어서의 난점

차륜형 도립진자 이동로봇은 단순히 자세제어만이 요구되던 링크형 도립진자의 경우와 달리 자세를 안정화시키는 동시에 요구되는 속도로 주행할 수 있도록 제어 해야하므로 다음과 같은 난점이 있다.

첫째, 자세 및 속도를 모두 정상편차 없이 제어하는 것은 불가능하다. 이 시스템은 입력의 수보다 출력 수가 많기 때문에 모든 출력을 정상편차가 생기지 않도록 제어하는 것은 불가능하다. 그래서 문제가 되는 것이 자세 및 속도제어간의 균형, 즉 자세와 속도 중 어느 쪽에 비중을 두고 제어기를 설계할 것인가 하는 것이다. 차륜형 도립진자 이동로봇은 항상 자세의 안정성이 확보되어야 하지만, 출발, 정지, 주행 등의 변속요구가 빈번하게 발생할 때는 주행제어시스템의 속도제어기능이 잘 동작되지 않으면 주어진 상황에 유연하게 대응할 수 없게 된다. 따라서 차륜형 도립진자 이동로봇의 자세 및 속도 제어기는 상위행동결정 레벨에서의 판단에 따라 주어지는 목표주행속도를 잘 추종하고, 또한 목표주행속도와 동특성에 의해 결정되어지는 목표자세의 근방에서 안정화하면서 주행할 수 있도록 설계되어져야 한다.

둘째, 자립형(self-contained)시스템에 주어지는 물리적 제약 내에서 최대의 성능을 발휘하는 제어시스템을 구성하여야 한다. 이동로봇과 같이 자신의 제어에 필요한 전원, 센서 시스템 등의 모든 기능을 내장하여야 하는 자립형 시스템의 경우, 전체시스템의 중량, 전원용량등을 고려하여 계산기가 결정되므로 어느 정도 계산능력에 한계가 있는 것이 탑재될 수가 있다. 따라서 차륜형 도립진자이동로

봇의 자세 및 속도제어에도 비선형 시스템의 제어에 적합한 퍼지이론이나 학습에 근거한 제어기법을 적용할 수 있으나, 가능한 한 복잡한 계산과정을 피하고, 단순하면서도 성능이 뛰어난 제어시스템을 구성하여야 한다는 어려움이 있다.

셋째, 자세검출용 자이로 센서의 드리프트 오차의 해결이다. 로봇을 비롯한 기타 다른 시스템의 자세검출에는 자이로 센서를 비롯하여 엔코드 센서, 포텐쇼미터, 가속도 센서들을 사용하고 있지만, 자이로센서에 비해 다른 센서들은 응답속도가 떨어지거나 정도면에서 뒤진다. 이러한 이유에서 자세 검출용 센서로 자이로 센서를 선호하고 있지만, 여기에는 온도드리프트에 의한 오차의 해결이라는 어려운 과제를 안게 된다. 자이로 드리프트오차의 보상에 관한 많은 연구가 시도된바 있으나, 획기적 연구결과는 아직 보고된 바 없다. 그러나 시스템의 동특성과 거의 유사한 선형모델만 정확히 구해진다면 상태관측기 등을 통해 자세는 추정될 수 있다.

이러한 관점에서 차륜형 도립진자 이동로봇의 자세 및 속도 제어를 위해 넓은 운전범위에서 시스템의 동특성과 유사한 특성을 가지는 최적의 선형모델을 도출하는 것은 의미있는 연구라 생각되어진다.

2.3 비선형 모델

본 절에서는 차륜형 도립진자 이동로봇의 운동방정식을 유도하고, 비선형 모델을 도출한다. 그림 2.2는 1차원의 차륜형 도립진자 이동로봇의 개략도를 나타낸 것이고 표2.1은 파라미터 및 변수들을 정의한 것이다. 여기서 파라미터의 값은 실제시스템을 참고로 결정되었

으며, 선형모델추정에 있어 비교대상이 되는 플랜트의 동특성을 결정하는 요소가 된다.

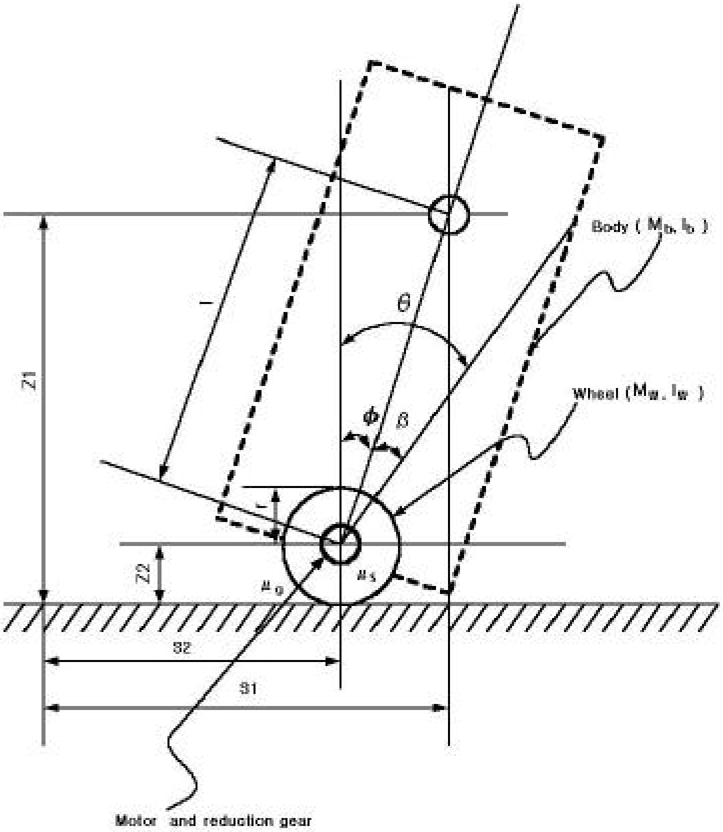


그림 2.2 **차륜형 도립진자 이동로봇의 개략도**

Fig. 2.2 Schematic diagram of the wheeled inverted pendulum type mobile robot

표 2.1 파라미터와 변수

Table 2.1 Parameters and Variables

파라미터	파라미터와 변수이름	값	단위
M_b	본체의 질량	12.18	[kg]
M_w	차륜의 질량	0.51	[kg]
I_b	본체의 회전관성	0.35	[kg · ㎡]
I_w	차륜의 회전관성	5.1E-4	[kg · ㎡]
I_M	모터축의 회전관성	3.2E-6	[kg · ㎡]
r	차륜의 반경	0.062	[m]
l	차축에서 중심까지의 거리	0.143	[m]
μ_s	차축의 점성마찰계수	5.76E-3	[Nm/(rad/sec)]
μ_g	접지면의 점성마찰계수	4.25E-3	[Nm/(rad/sec)]
τ_t	모터의 토크 정수	23.5E-3	[Nm/A]
g	중력가속도	9.8	[m/sec^2]
η	차륜의 최종감속비	39.5	
ϕ	본체의 기울기각		[rad]
θ	차륜의 회전각		[rad]
β	본체의 기울기각과 차륜의 회전각의차		[rad]
u	모터의 전기자 전류		[A]

로봇의 운동에너지 T, 위치에너지 U, 손실에너지 D, β 축에 대한 외력 Q_β, θ 축에 대한 외력 Q_θ 라 하면 이들은 각각 그림 2.2로부터 다음과 같이 주어진다.

$$T = \frac{1}{2}M_w(\dot{s}_2^{\,2} + \dot{z}_2^{\,2}) + \frac{1}{2}M_b(\dot{s}_1^{\,2} + \dot{z}_1^{\,2}) + \frac{1}{2}I_w\dot{\theta}^2 + \frac{1}{2}I_b(\dot{\theta} - \dot{\beta})^2 + \frac{1}{2}I_M\eta^2\dot{\beta}^2 \tag{2.1}$$

$$U = M_w gr + M_b gl\cos(\theta - \beta) \tag{2.2}$$

$$D = \frac{1}{2}(\mu_s\dot{\beta}^2 + \mu_g\dot{\theta}^2) \tag{2.3}$$

$$Q_\beta = \eta\tau_t u \tag{2.4}$$

$$Q_\theta = 0 \tag{2.5}$$

차륜형 도립진자 이동로봇의 운동방정식을 도출하기 위해 β축과 θ축을 일반화 좌표로 하는 Lagrange 운동방정식을 나타내면 다음과 같다.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\beta}}\right) - \frac{\partial L}{\partial\beta} + \frac{\partial D}{\partial\dot{\beta}} = Q_\beta \tag{2.6}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}}\right) - \frac{\partial L}{\partial\theta} + \frac{\partial D}{\partial\dot{\theta}} = Q_\theta \tag{2.7}$$

단, 여기에서 L은 라그랑지안(Lagrangian)으로서 L=T-U로 구할 수가 있다.

식 (2.6)과 (2.7)에 식 (2.1) (2.5)를 대입하여 θ와 ϕ 에 관하여 정리하면, 식 (2.8), 식 (2.9)의 차륜형 도립진자 이동로봇의 비선형 운동방정식을 얻을 수 있다.

$$(M_b l^2 + I_b + \eta^2 I_M)\ddot{\phi} + (M_b r l\cos\phi - \eta^2 I_M)\ddot{\theta} + \mu_s\dot{\phi} - \mu_s\dot{\theta} - M_b g l\sin\phi = -\eta\tau_t u \tag{2.8}$$

$$(M_b r l\cos\phi + M_b l^2 + I_b)\ddot{\phi} + [(M_b + M_w)r^2 + M_b r l\cos\phi + I_w]\ddot{\theta} - M_b r l\,\ddot{\phi}^2\sin\phi + \mu_g\dot{\theta} - M_b g l\sin\phi = 0 \tag{2.9}$$

따라서 플랜트의 상태변수를 $\mathbf{x}_p = [x_{p1}\ x_{p2}\ x_{p3}]^T = [\phi\ \dot{\phi}\ \dot{\theta}]^T$라 정의하면 비선형 상태방정식은 다음과 같이 나타낼 수 있다.

$$\dot{x}_p = f(x_p, u, t) \tag{2.10}$$

제 3장 SA를 접목한 RCGA기반 모델 선형화

본 논문에서는 차륜형 도립진자 이동로봇의 자세 및 속도제어를 위해 넓은 동작범위에 있어서 식 (2.10)으로 표현되는 플랜트의 동특성과 유사한 특성을 가진 선형모델을 도출하는 것을 목적으로 한다. 선형화 모델은 동작점 부근에서 Taylor급수 전개함으로써 얻어질 수 있으나, 이에 근거하여 제어기를 설계할 경우 운전범위가 지극히 제한되어질 수 있다. 따라서 보다 넓은 동작범위에 걸쳐 플랜트와 유사한 동특성을 갖는 최적선형 모델 도출을 위한 노력이 필요하다.

본 장에서는 최근 최적화 기법 중 자주 이용되고 있는 실수코딩 유전알고리즘(Real-Coded Genetic Algorithm: RCGA)과 시뮬레이티드어닐링(Simulated Annealing: SA)에 대해 언급하고, 본 논문에서 제안하는 SA를 접목한 RCGA기반 모델선형화에 대해 기술한다.

3.1 RCGA

찰스다윈(Charles Darwin, 1809 1882)은 1895년 발행된 "종의 기원"에서 종이 어떻게 변할 수 있는지를 설명하는 메커니즘으로 자연선택을 제시하였다. 다윈은 자연에 잘 적응하는 개체는 생존하고 그렇지 못하면 도태 당하는 “적자생존의 원리”, 즉 자연에 의해 선택된다는 원리로 생물학적 진화를 설명하였다.

유전알고리즘은 자연선택과 유전법칙을 모방한 확률적 탐색기법이다. 이 알고리즘은 1975년 Holland에 의해서 처음 소개되었다[11]. 유

전알고리즘의 가장 큰 특징은 뉴톤법과 같은 고전적 최적화기법이나 타부서치(Tabu search), 시뮬레이티드어닐링과 같은 이웃 탐색기법이 하나의 해를 운용하는 데 반하여, 유전알고리즘은 복수개의 잠재 해들로 이루어진 해의 집단(population)을 운용한다는 것이다. 이러한 해 집단에 자연선택과 유전법칙의 메커니즘을 적용하여 세대를 진행시키면서 해공간을 탐색해간다. 그림 3.1은 유전알고리즘의 흐름도를 나타낸 것이다.

유전알고리즘은 좋은 해의 이용만을 강조하면 조기 수렴하여 부분최적에 빠질 수 있고 해 공간의 탐색만을 강조하며 임의탐색에 가까워 좋은 해를 찾아가지 못하게 된다. 유전알고리즘은 매 세대마다 모집단을 운용함으로써 내재된 병렬성을 갖는 기법이다. 전통적으로 유전알고리즘의 표현법은 이진코딩이었다. 이진코딩은 광범위한 분야에서 폭 넓게 채용되고 있지만 탐색되는 정의 영역이 크거나, 고정밀도의 해가 요구되거나, 제약조건(구속조건)이 따르는 최적화 문제에는 적용하기가 어렵다. 이진 코딩의 이러한 제약성을 해결해 줄 수 있는 방법이 실수코딩이다. 실수코딩을 채용함으로써 염색체는 해 벡터 요소들을 부호화 필요 없이 이들로 직접 표현할 수 있다[8].

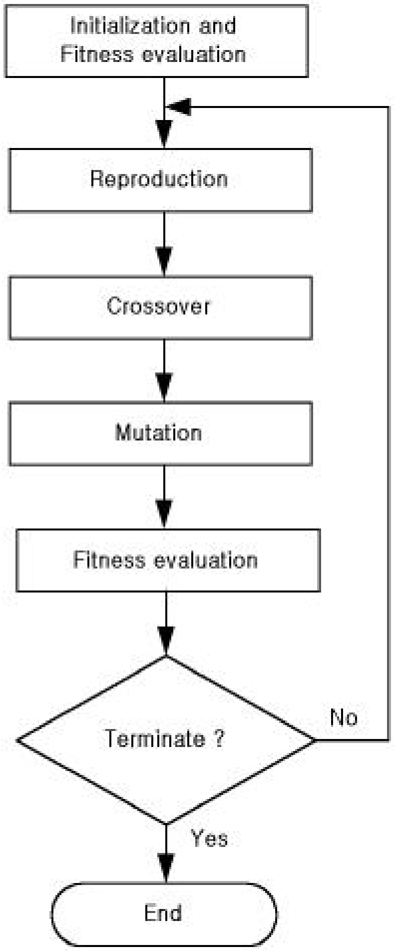


그림 3.1 **유전알고리즘의 흐름**

Fig. 3.1 Flow chart of a GA

3.1.1 염색체 표현

기존의 이진코딩의 문제점을 해결하기 위한 한 방법이 실수코딩이다. 실수코딩을 채용함으로써 염색체는 해 벡터의 요소들로 구성된다. k세대의 실수코딩 염색체는 식 (3.1)과 같이 표현된다.

$$s(k) = x^T(k) = (x_1(k)\ x_2(k) \cdot \cdot \cdot \cdot \cdot\ x_n(k)) \tag{3.1}$$

여기서 $x(k) \in \mathbb{R}^n$는 해 벡터이고 n 은 벡터의 차원이다. 실수코딩을 채용함으로써 염색체의 유전자와 벡터요소는 정확히 일대일 일치하는 관계를 가지며 염색체의 길이는 벡터의 차원 n과 같게 된다[8]. 그림 3.2는 실수코딩 염색체의 한 예를 나타낸 것이다.

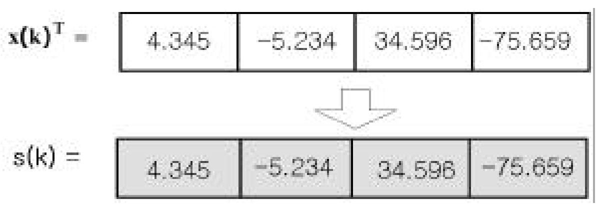


그림 3.2 실수코딩 염색체

Fig. 3.2 Real coding chromosome

3.1.2 초기 집단의 생성

N개의 염색체 개체로 구성되는 k세대의 집단은 다음과 같이 표시된다.

$$p(k) = \{s_1(k)\ s_2(k) \cdot \cdot \cdot \cdot \cdot s_N(k)\} \tag{3.2}$$

여기서 $s_i(k) = (x_{i1}(k)\ x_{i2}(k) \cdot\cdot\cdot x_{ij}(k) \cdot\cdot\cdot x_{in}(k))$는 i번째 염색체, $x_{ij}(k)$는 i번째 요소, N은 집단의 크기이고, 이는 세대와는 관계없이 고정되는 것으로 간주한다. 염색체 요소들은 난수발생기를 통해 발생되는 정의영역 $x_j^L \le x_{ij}(k) \le x_j^U\ (1 \le i \le N, 1 \le j \le n)$ 내의 실수 값으로 설정한다[8].

3.1.3 재생산 연산자

적자생존(자연도태)의 현상을 모방하여 각 개체의 적합도를 평가하는 기구를 도입하고, 적합도에 따라 개체를 선택하고 새로운 집단을 형성하게 되는데 이를 재생산이라 한다. 재생산은 적합도의 크기에 따라 개체들을 선택하기 때문에 적합도가 낮은(열성의 유전인자를 포함) 개체들은 집단에서 소멸하게 되는데 이로서 전체 집단의 평균 적합도는 증가하게 되고 개체들은 점점 해에 접근하게 된다.

재생산을 알고리즘 형태로 구현한 것으로는 루울렛휠 선택에 기초한 재생산, 토너먼트 선택에 기초한 재생산, 구배와 유사한 재생산 등의 여러 방법들이 있다[8]. 본 논문에서는 구배와 유사한 재생산을 사용한다.

3.1.4 교배

자연계 생물들의 개체 증식은 부모 세대의 성적결합 즉, 교배를 통해 이행되면 이를 통해 서로간의 유전인자를 교환하게 된다. 유전알고리즘에서도 개체간의 정보교환을 통해 새로운 점을 탐색하기 위

하여 자연계의 교배현상을 이용한다. 자연계의 교배를 알고리즘 형태로 흉내낸 교배 연산자로는 단순교배, 산술 교배, 수정단순교배, 프랫교배, 선형교배 등이 있으나 본 논문에서는 아래와 같이 기술되는 산술교배를 사용한다.

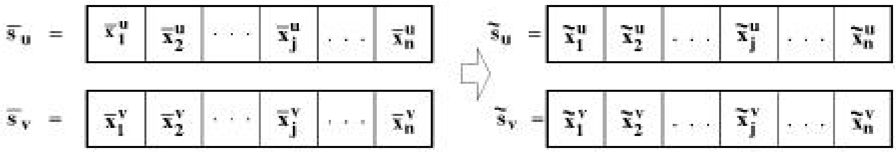


그림 3.3 산술 교배

Fig. 3.3 Arithmetical crossover

$$\tilde{x}_j^{u} = \lambda \ \bar{x}_j^{v} + (1-\lambda) \ \bar{x}_j^{u} \qquad (3.3)$$

$$\tilde{x}_j^{v} = \lambda \ \bar{x}_j^{u} + (1-\lambda) \ \bar{x}_j^{v} \quad (1 \le j \le n) \qquad (3.4)$$

여기서 λ는 곱인수로서 고정되거나 아니면 각 요소마다, 염색체마다 독립적으로 결정되며, 0 1사이 값을 가진다[8].

3.1.5 돌연변이

자연계에서 부모 세대가 보유한 유전자가 특성 이외에 특이한 변이가 발생하는 것을 돌연변이라 하며, 이러한 변이의 발생은 과거의 부모세대에서 보유하였던 특성과는 다른 경향으로 그 개체적 특성을 보인다. 돌연변이는 유전적 기형을 만들고, 그것을 영속시키기 때문에 위험스럽기도 하지만 경우에 따라서는 유익한 방향으로 작용하기도 한다. 재생산과 교배는 적합도에 따라 개체들을 선택하고

이들간의 정보교환이 가능케 함으로서 집단을 강하게 해 주지만, 염색체들이 특정 비트(또는 요소)가 모두 같게 되면(특정위치 요소가 같은 값을 가짐)이를 변경하는 것이 어려울 때가 있다. 이러한 현상은 세대 말기에는 바람직하지만 세대초기에 발생하게 되면 유전알고리즘이 지역해나 사점에 빠지게 되는 원인이 된다. 이러한 원치않는 해 쪽으로 조기 수렴하는 것을 막기 위해서 도입한 것이 돌연변이이다.

돌연변이 연산자에는 균등돌연변이, 경계돌연변이, 동적 돌연변이를 포함한 여러 가지가 있다. 여기서는 본 논문에서 사용된 동적 돌이변이에 대하여 기술한다. 일명 불균등 돌연변이라고도 불리우는 이 연산자는 정밀도를 높이기 위하여 미세조정이 가능하도록 고안된 것이다. 어떤 j 번째 유전자에서 돌연변이가 일어나면 x_j는 식 (3.5)와 같이 결정된다.

$$x_j = \begin{cases} \tilde{x}_j + \Delta(k,\ x_j^{(U)} - \tilde{x}_j), & \tau = 0\text{일때} \\ \tilde{x}_j - \Delta(k,\ \tilde{x}_j - x_j^{(L)}), & \tau = 1\text{일때} \end{cases} \qquad (3.5)$$

여기서 τ는 0 또는 1 둘 중에서 하나를 취하는 난수이다. $\Delta(k,y)$에는 식 (3.6)이 이용될 수 있다.

$$\Delta(k,y) = y \cdot r \cdot \left(1 - \frac{k}{T}\right)^b \qquad (3.6)$$

여기서 r 은 0과 1 사이의 실수난수이고, T는 알고리즘의 실행되는 최대 세대수이고, b는 불균등 정도를 나타내는 매개변수로서 사용자

에 의해 결정된다. 이 함수는 영역 [0,y]에서 값을 가지는데, 세대수 k 가 증가함에 따라 0에 가까운 수를 발생할 확률이 높아지도록 접근하게 된다. 이 연산자의 특징은 초기에는 전 공간을 균등한 확률로 탐색하다가 세대수가 증가하면 매우 지역적으로 탐색하도록 해 준다[8].

3.1.6 적합도 평가

적합도의 개념은 다윈의 자연도태설을 근간으로 하는데, 자연계의 생물들이 처한 환경에 적응해 가는 생존능력을 말한다. 유전알고리즘에서도 인위적인 집단내에서 각 개체의 적응능력은 측정하는 메커니즘으로 적합도를 사용하면 이는 목적함수로부터 계산된다. 적합도가 환경에 대한 개체의 장점을 의미하듯이, 유전알고리즘에서도 적합도 값은 항상 양의 값을 가지도록 할 필요가 있다.

최대/최소화 문제의 형태로 기술되는 임의의 목적함수에 대해 적합도 함수는 여러 가지 방법으로 계산될 수 있지만, 다음과 같은 사상을 통해 얻을 수 있다.

최대화: $f(s(k)) = F(x(k)) - r$

최소화: $f(s(k)) = -F(x(k)) - r$

여기서 $f(s(k))$는 적합도 함수, $F(x(k))$는 목적함수이고, r은 항상 $f(s(k)) \geq 0$의 관계가 유지되도록 선정되는 상수이다. r은 유전알고리즘의 성능에 직접적인 영향을 미치는 요소로서 부적절하게 선정되면 탐색의 효율이 떨어진다[8].

3.2 SA

SA의 원리는 반복적 개선법을 기본으로 하되 비용증가의 이동을 확률적으로 허용하는데 있는데 Kirkpatrick등에 의해 1983년도에 최초로 소개되었다[16]. 그 원리는 고체물리학에서 에너지 수준이 가장 낮은 상태인 결정을 얻기 위해서 이용하고 있는 어닐링 과정을 그대로 조합적 최적화 문제에 흉내낸 것이다. 고체를 열탕에 넣고 고체가 녹아서 액체상태가 될 때까지 가열한 후 열탕의 온도를 서서히 낮추어 가장 안정된 결정상태의 고체를 얻는 과정을 고체물리학에서 어닐링(annealing)이라고 부른다. 액체상태에서는 분자들이 자유롭게 움직이다가 온도를 내림에 따라 점점 일정한 구조로 배열하게 되는데, 온도를 급격히 내리면 가변적 안정상태의 비정질의 구조가 얻어지게 된다. 이 때 물질의 각 상태마다 에너지 수준이 다른 값을 갖게 되는데 물리적인 구조를 조합최적화 문제에서의 하나의 가능해의 구조로 대응시키면 가장 낮은 에너지 수준을 주는 상태에 대응하는 것이 바로 최적해가 된다. 온도를 급격히 내려 에너지 수준을 급히 내리는 quenching에 해당하는 것이 바로 지역 탐색법이고, 서서히 내리면서 분자들이 배열을 바꾸면서 에너지 수준을 오르내리도록 허용하는 어닐링에 대응하는 것이 시뮬레이티드 어닐링 알고리즘이 된다. 즉, 물리적 상태 변화 과정을 조합적 최적화문제와 연계시키면 표 3.1과 같이 요약할 수 있다.

표 3.1 물리적 어닐링과 SA

Table 3.1 Physical annealing and SA

물리적 어닐링	SA
물질	조합적 최적화 문제
가능한 상태	가능 해 (feasible solution)
에너지	비용함수
기저 상태	최적해
어닐링	시뮬레이티드 어닐링
quenching	지역탐색법

문제는 어닐링을 어떻게 확률적으로 모사할 것인가 라는 것이다. 이것의 기본형태는 통계역학 분야에서 이미 1953년에 Metropolis 알고리즘으로 개발되었다. Metropolis 알고리즘은 열탕에서 고체가 열평형상태(thermal equilibrium)에 도달하는 과정의 확률적 시뮬레이티드 방법으로 그 과정은 우선 고체의 초기상태를 무작위적으로 선택한 후 현재상태 x에서 약간의 변동(perturbation)을 주어 다음의 상태 y를 만들고 에너지의 차이 $\Delta = E(y) - E(x)$가 0 이하이면 s′으로의 변동을 받아들이고 0 보다 크면 s′으로의 변동을 식 (3.7)과 같은 확률로 받아들여 현재상태를 변경시키는 과정을 반복한다.

$$\exp\left(-\frac{\Delta}{k_B T}\right),\quad k_B : \text{볼쯔만 상수},\ T : \text{열탕의 온도} \qquad (3.7)$$

온도를 고정시키고 이 과정을 충분히 많이 반복하면 결국은 고체

의 상태분포가 식 (3.8)과 같은 볼쯔만분포(Boltzmann distribution)를 따르게 되어 열평형에 도달함을 알 수 있다.

$$P_T\{s = x\} = \frac{1}{Z(T)} \exp\left(\frac{-E(x)}{k_B T}\right) \qquad (3.8)$$

여기서 Z(T)는 식 (3.8)을 모든 가능한 상태에 대해 계산하여 모두 더했을 때 합이 1이 되도록 하는 상수로 분할함수(partition function)이다.

시뮬레이티드 어닐링은 Metropolis 알고리듬을 기본으로 하여 어닐링에서의 온도를 낮추는 과정을 덧 붙여서 얻은 방법으로 볼 수 있는데, 이 때 온도를 알고리즘의 제어파라미터의 역할을 하게 된다. 현재 실용적으로 사용되어지고 있는 시뮬레이티드 어닐링 알고리즘의 기본 형태는 그림 3.4에 나타낸다.

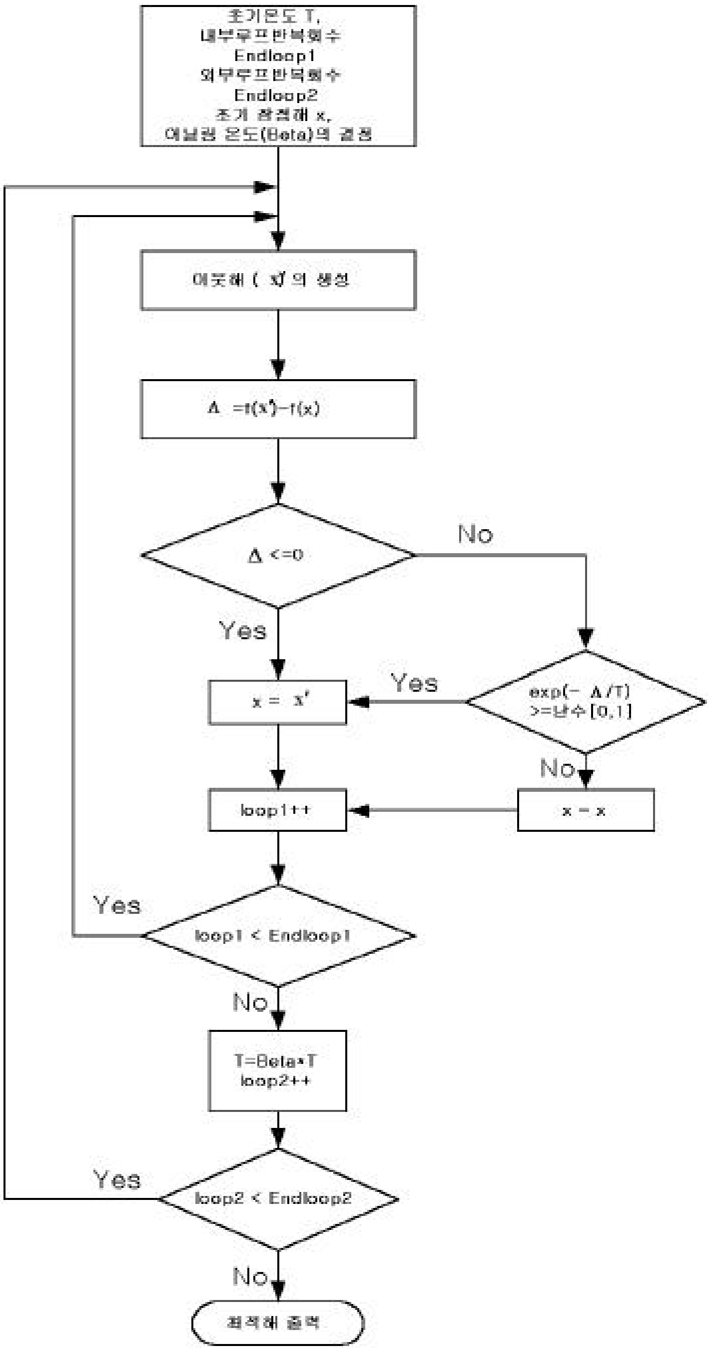


그림 3.4 SA**의 흐름도**

Fig. 3.4 Flow chart of a SA

3.2.1 초기 온도설정

SA에 있어서 초기온도는 이론적으로는 모든 이동을 받아들일 수 있도록 높게 설정해야 한다. 하지만 초기온도를 높게 설정하게 되면 지역 최소값에서 벗어나기는 쉽지만 알고리즘의 수행속도가 상당히 느려지게 된다. 반면 초기온도를 낮게 설정하게 될 경우에는 지역최소값에 빠질 경우가 있으므로 적절한 선택이 필요하다.

3.2.2 온도 감소계수의 설정

이론적으로는 온도를 내리는 함수는 감소함수로서 시간이 경과함에 따라 0으로 수렴해야 한다. 하지만 온도를 너무 급속히 낮추면 평형상태를 이루어서 최소에너지 상태에 도달할 확률이 적고, 너무 천천히 낮추면 최소에너지에 도달할 확률은 커지지만 많은 반복을 필요로 하므로 시간이 많이 걸리게 된다. 현재 대표적인 온도감소계수의 설정방법에는 로그스케쥴과 기하스케쥴을 이용하고 있다. 로그스케쥴의 경우에는 k번째 온도(T_k)를 $T_k = \frac{T_o}{\log(k)} (k \to \infty)$로 설정할 때 확률적으로 최적해에 도달하는 것이 증명되어있지만 실제문제에 있어서 잘 사용되지 않는다. 현재 자주 사용되는 것이 기하스케쥴이며, $T_k = \gamma \times T_k$로 설정되며 γ는 경험적으로 $0.8 \le \gamma < 1.0$로 설정되는 경우가 많다.

3.2.3 종료조건

SA에 난점중의 하나가 적절한 종료시점의 판정이다. 종료기준은 알고리즘이 끝나는 조건으로서 이론적으로는 온도가 0에 수렴하는 곳에서 끝나야 하지만 알고리즘이 낮은 온도에서 보내는 시간이 너무 많기 때문에 보통 외부루프가 일정회수 반복할 때까지 해의 향상이 없으면 종료하는 방식을 취하고 있다.

3.3 SA를 접목한 RCGA

RCGA는 임의의 함수를 최적화하는 데에 있어서 저 차원에서는 아주 유용한 방법이다. 그러나 VLSI의 CAD 문제 등과 같은 복잡한 문제에 있어서는 SA보다 덜 이용되고 있다. RCGA의 경우 교배(crossover)에 대한 의존도 상당히 높다. GA에서 개체가 동형이 되면 사점이나 지역해에 빠질 수 있는데 재생산과 교배는 이로부터 벗어나게 할 수 없다. 이런 문제점을 해결하기 위해서 돌연변이연산자가 필요하지만, 일반적으로 돌연변이 확률은 낮게 설정되기 때문에 탐색공간이 적다. 또한 지역탐색에 매우 약한 편이다[10, 15].

따라서, RCGA의 단점을 보완하기 위하여 지역탐색하는 장점을 가진 SA를 결합하는 것이 필요하다. 모든 세대의 어떤 점에서 더 높은 적합도를 가진 개체가 존재하면 다음세대의 진행 전에 더 좋은 개체를 생산할 수 있다. 이것은 그림 3.5에 나타낸 바와 같이 일반적인 RCGA에 SA를 결합함으로써 이러한 문제점의 해결이 가능할 것이다.

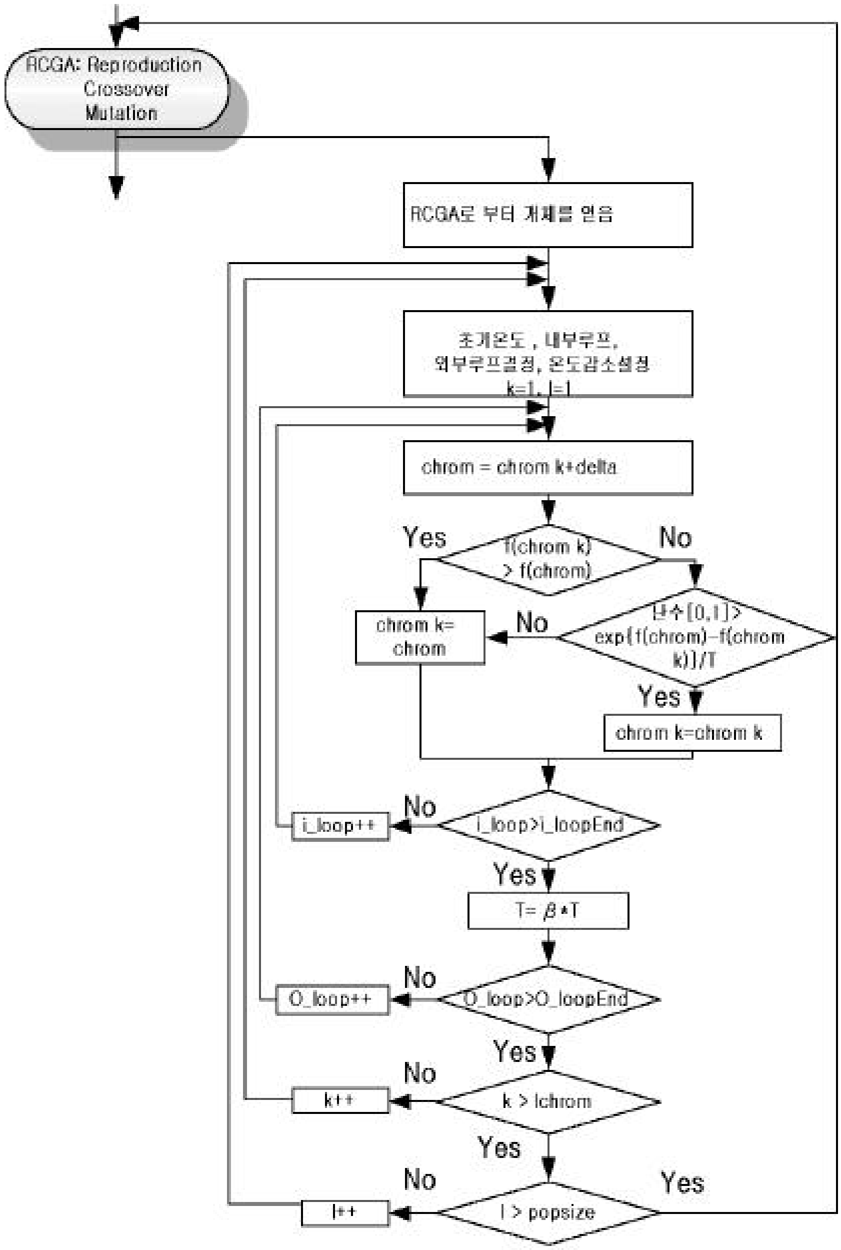


그림 3.5 SA**를 접목한** RCGA**의 흐름도**

Fig. 3.5 Folw chart of the RCGA with SA

3.4 선형모델의 최적 파라미터 추정

식 (2.10)으로 표현되는 시스템은 비선형 요소를 내포하고 있기 때문에 선형제어기를 설계하기 위해서는 비선형 시스템과 유사한 특성을 가지는 선형화 모델이 요구된다. 식 (2.10)의 시스템을 직립자세(ϕ=0, $\dot{\phi}$=0) 근방에서 선형화하고, $\mathbf{x}_m = [x_{m1}\ x_{m2}\ x_{m3}]^T = [\phi\ \dot{\phi}\ \dot{\theta}]^T$ 라 정의하면 식 (3.9)와 같은 선형상태방정식을 얻을 수 있다.

$$\dot{\mathbf{x}}_m(t) = A x_m(t) + B u(t) \tag{3.9}$$

단, $A = \begin{pmatrix} 0 & 1 & 0 \\ a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ b_1 \\ b_2 \end{pmatrix}$ 이다.

이러한 방법으로 얻어진 모델에 근거하여 로봇의 자세 및 속도제어기를 설계할 경우, 직립자세부근에서의 자세안정화가 보장될 수 있으나, 고속 주행시 요구되는 속도제어성능이 저하될 수 있다. 따라서 본 논문에서는 직립자세에서부터 정상주행속도를 가능한 한 넓은 운전범위에 걸쳐서 대상시스템의 동특성과 유사한 특성을 갖는 식 (3.9)로 표현되는 시스템 파라미터 a_1, a_2, a_3, a_4, a_5, a_6, b_1, b_2를 모델 조정기법과 SA를 접목한 RCGA를 이용하여 추정한다.

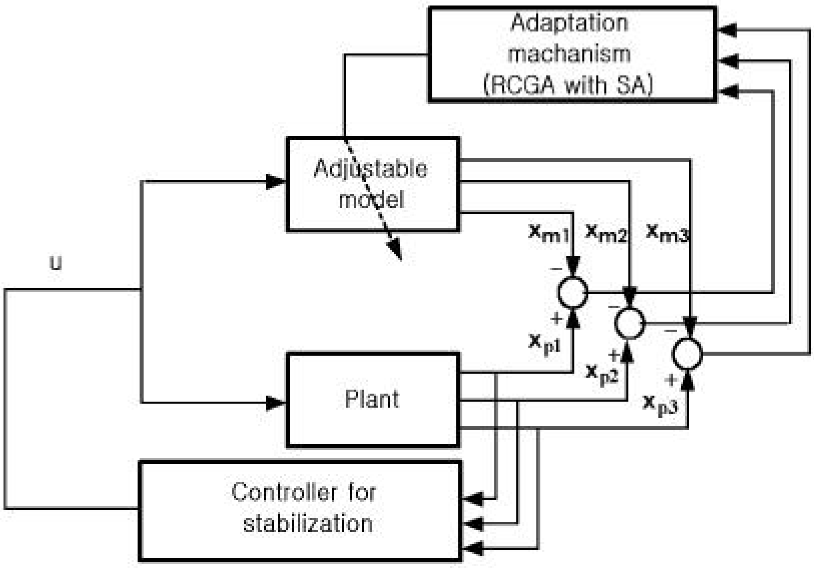


그림 3.6 SA를 접목한 RCGA를 이용한 대상시스템의 파라미터 추정
Fig. 3.6 Parameters estimation of the controlled system
using a RCGA with SA

그림 3.6은 제안된 방법에 의해 선형모델의 파라미터를 추정하는 구조를 나타낸 것이다.

대상 시스템과 선형화 될 모델은 병렬로 연결되어져 있으며, 동일한 입력 u를 부가한 후 각각의 출력으로부터 다음과 같은 목적함수가 계산되어 진다.

$$J = \int_0^t [x_p - x_m]^T P [x_p - x_m] dt \tag{3.10}$$

여기에서 x_p 는 실제시스템의 상태벡터이고, x_m 은 모델의 상태벡터를 의미한다. 또한 P는 하중행렬로서 $P = diag(P_{11}, P_{22}, P_{33})$

이다.

식 (3.9)의 선형시스템의 파라미터는 식 (3.10)의 목적함수가 최소가 되도록 SA를 접목한 RCGA 적응메커니즘에 의하여 최적으로 조정된다. 또한 대상시스템이 불안정한 시스템이므로 임의의 입력 u가 주어지면 시스템이 발산하게 되므로 파라미터추정에 이용되는 입력 u는 임의의 안정화용 제어기를 통해서 결정되어진다. 파라미터 추정 시 대상시스템을 안정화하기 위한 목적으로 이용된 제어기구조로서 제어입력 u는 다음과 같이 나타내어진다.

$$u = -k_1 x_{p1} - k_2 x_{p2} - k_3 x_{p3} \qquad (3.11)$$

단, $[\,k_1\; k_2\; k_3\,]=[17.7\;\; 3.04\;\; 0.181]$

그림 3.7은 그림 3.5로 표현되는 대상시스템의 초기값을 $x(0) = [5\pi/180\;\; 0\;\; 0]$로 설정했을 때 식 (3.11)으로 표현되는 안정화용 제어기에 의해 결정된 입력 u를 나타낸 것이다. 그림 3.8과 그림 3.9는 각각 그림 3.7의 u가 주어졌을 때 적응자를 RCGA로 사용했을 경우와 SA를 접목한 RCGA를 사용했을 때의 각 파라미터가 수렴해 가는 과정을 나타낸다. 또한 그림 3.10은 이 때의 목적함수의 추이를 나타낸 것이다. 파라미터 추정 시 이용된 기타 초기조건은 다음과 같다.

하중행렬 $P = \mathrm{diag}(220,\; 3.0,\; 2.4)$
세대수 : 300
집단수 : 40
교배확률 : 0.95
돌연변이확률 : 0.1
초기온도 : 5
온도감소계수 : 0.95
내부루프 : 5
외부루프 : 5

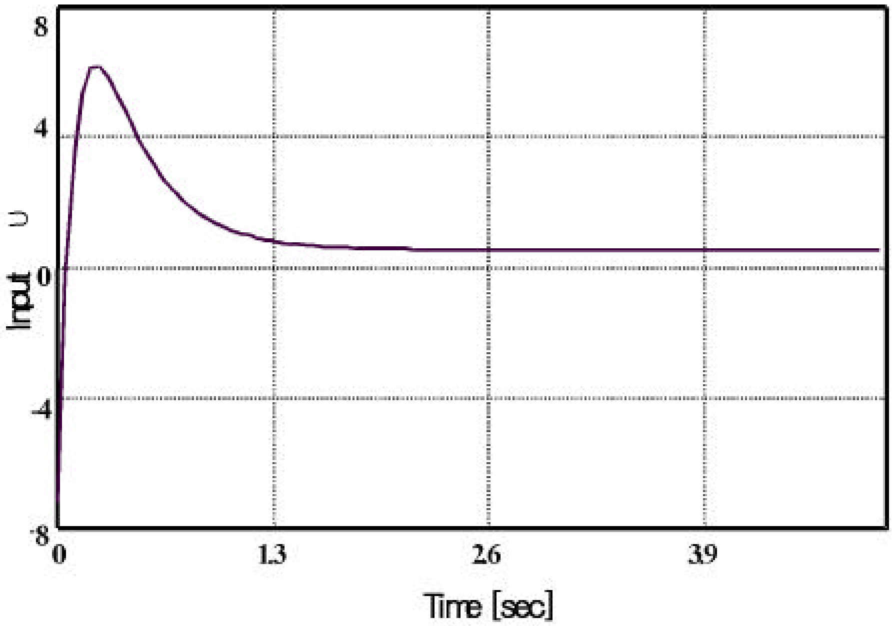
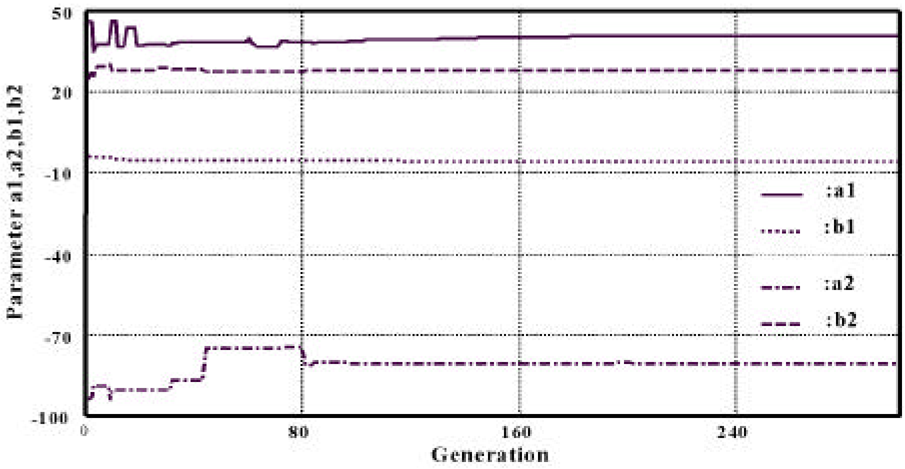
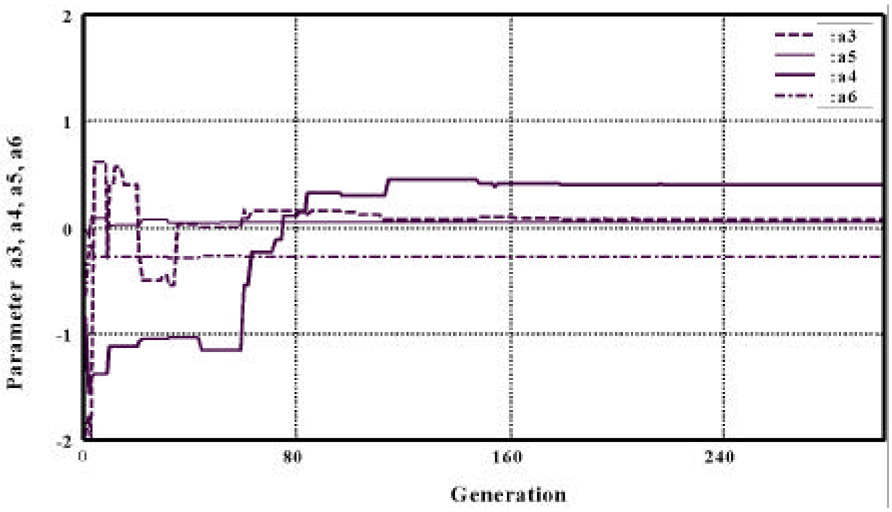


그림 3.7 파라미터 추정을 위한 입력 데이터

Fig. 3.7 Input data for parameter estimation



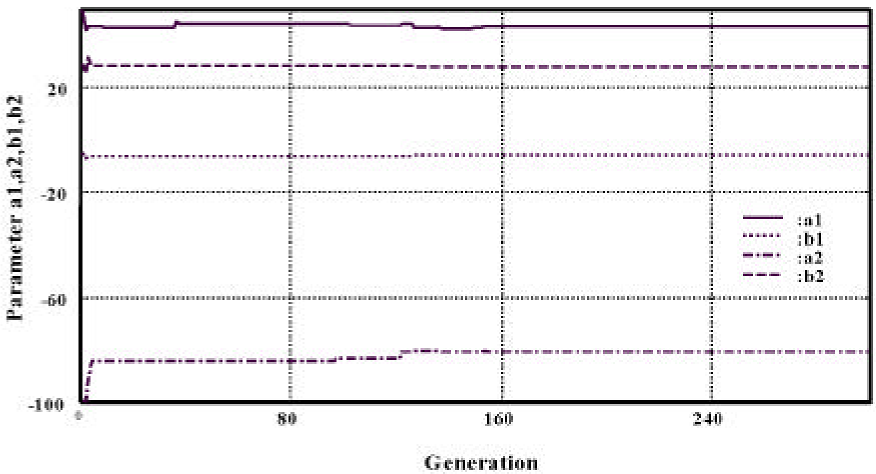
(a) Parameters (a1, a2, b1, b2)



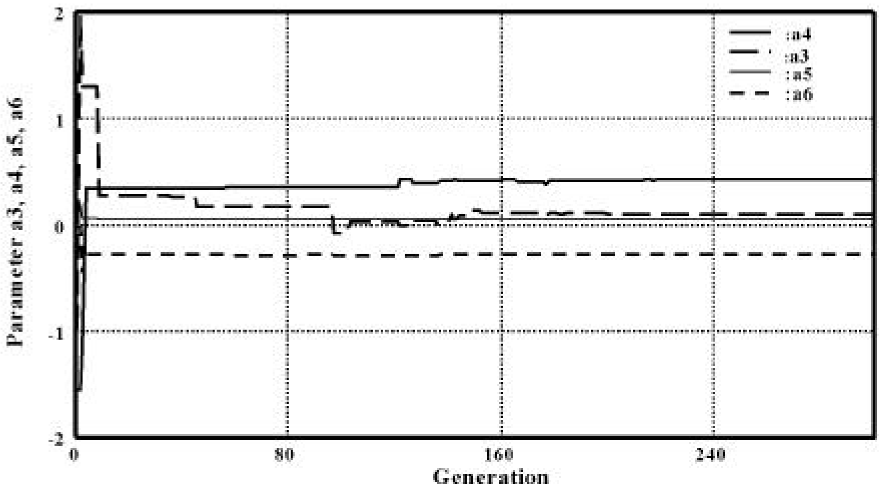
(b) Parameters (a3, a4, a5, a6)

그림 3.8 RCGA**를 이용한 파라미터의 추정결과**

Fig. 3.8 Results of parameter estimation using a RCGA



(a) Parameters (a1, a2, b1, b2)



(b) Parameters (a3, a4, a5, a6)

그림 3.9 제안한 방법을 이용한 파라미터의 추정결과

Fig. 3.9 Results of parameter estimation using the proposed method

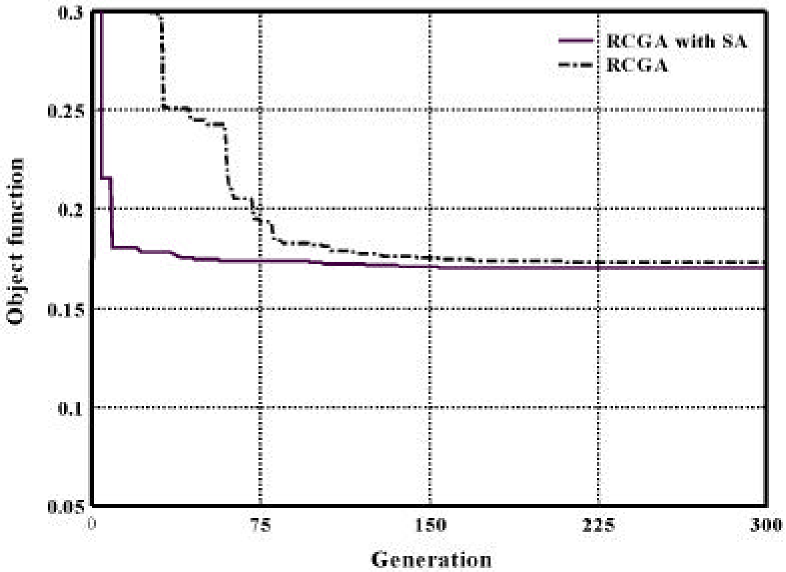


그림 3.10 RCGA **와 제안한 방법의 목적함수 비교**

Fig. 3.10 Comparison of the objective function between a RCGA and the proposed method

그림 3.8·9·10으로부터 제안된 방법에 의한 파라미터 추정의 경우가 RCGA에 의한 방법에 비해 적은 세대수에서 수렴하고 있음을 알 수 있다. 표 3.2는 RCGA와 제안된 방법에 의해 추정된 시스템 파라미터의 값을 나타낸다.

표 3.2 RCGA와 제안한 알고리즘에 의해 추정된 파라미터 값
Table 3.2 Parameter values estimated by RCGA and the proposed algorithm

적응자 / 시스템 행렬	RCGA	제안된 알고리즘
A	$\begin{bmatrix} 0 & 1 & 0 \\ 40.8195 & 0.0796 & 0.0522 \\ -80.4474 & 0.4082 & -0.2789 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 43.5915 & 0.0955 & 0.0531 \\ -80.7274 & 0.4259 & -0.2790 \end{bmatrix}$
B	$\begin{bmatrix} 0 \\ -5.8305 \\ 28.1734 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -6.076 \\ 28.1985 \end{bmatrix}$

제 4장 검증 시뮬레이션

4.1 시뮬레이션 조건

본 논문에서는 제안된 방법에 대한 유효성을 검증하기 위해 제안된 방법, RCGA 및 테일러 급수전개에 의해 얻어진 각각의 선형모델과 실제시스템에 동일한 입력을 부가하여 응답특성을 비교하기로 한다.

시뮬레이션에 이용된 프로그램은 Matlab에 의해 작성되었으며, 샘플링 시간은 0.05 sec로 주어졌다. 또한 초기 설정치는 본체의 기울기 각도를 $5\pi/180$, 차륜의 회전각을 0, 차륜의 회전 각속도를 0으로 하였다.

4.2 결과 및 검토

그림 4.1은 전술한 각 선형모델과 실제시스템에 그림 3.7과 같은 u를 인가했을 때의 응답을 나타낸 것으로 테일러 급수전개 방법이나 RCGA에 의해 얻어진 선형모델에 비해 본 논문에서 제안한 방법에 의해 얻어진 선형모델의 경우가 실제 시스템의 응답과 유사함을 알 수 있다.

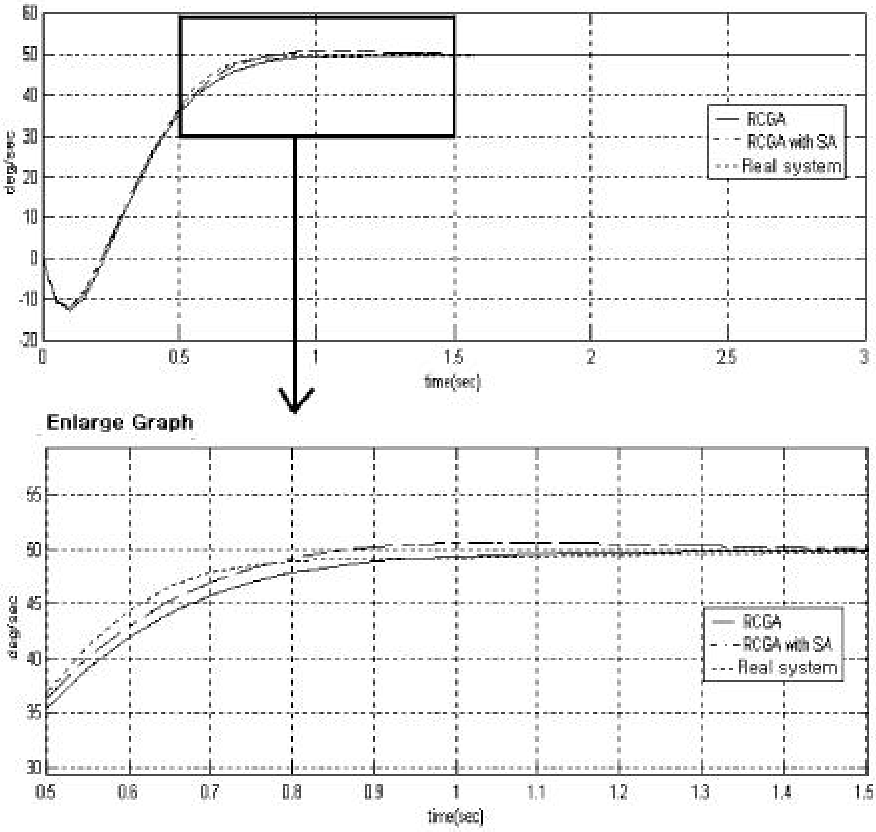


그림 4.1 **입력** u **에 대한 응답**

Fig. 4.1 Responses of input u

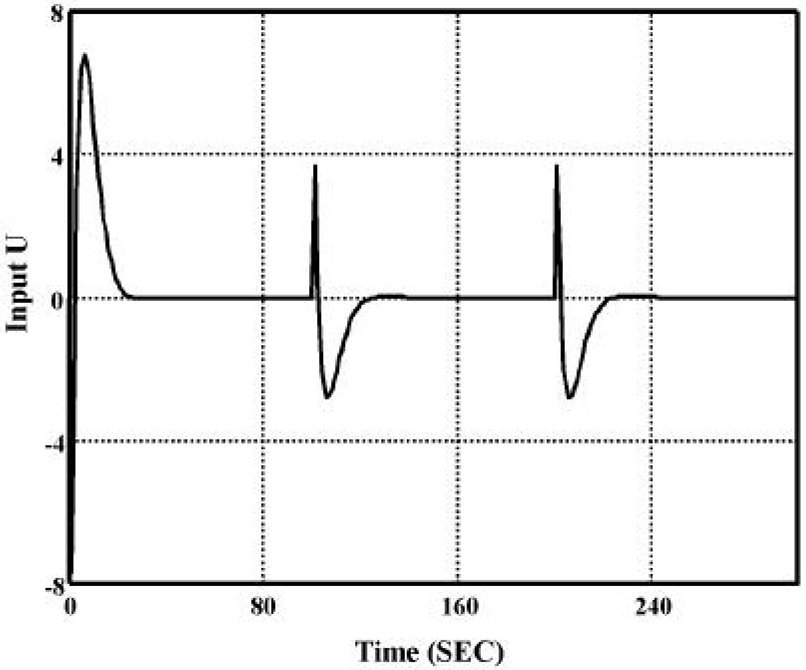


그림 4.2 **제어입력**

Fig. 4.2 Control input

다음으로 제어입력이 빈번하게 변화할 때의 응답을 비교하기 위하여 그림 4.2와 같은 제어입력을 인가하였다. 그림 4.3은 이 때의 응답을 나타낸 것으로 시간이 경과함에 따라 각 시스템의 응답이 거의 일치하고 있으나 과도응답 특성을 살펴보면 제안된 알고리즘에 의해 추정된 선형모델의 응답이 실제시스템의 것과 가장 유사함을 알 수가 있다.

이러한 결과로부터 제안된 방법에 의한 타당성을 검증할 수 있었다. 이러한 결과는 실험에 의해 더욱 분명히 나타날 것으로 생각된다. 왜냐하면 시뮬레이션에서는 실제시스템 대신에 식 (2.10)의 비선형 시스템을 이용하는 것으로서 차축 또는 차륜과 접지면에서의 점성 마찰계수등이 주어졌지만 실제로 이 값들은 측정하는 것이 쉽지 않다. 따라서 실제시스템을 이용한 실험을 통해서 선형모델을 구하는데 제안된 알고리즘이 적용되면 보다 더 실제시스템에 근접한 모델을 구할 수 있으리라 사료된다.

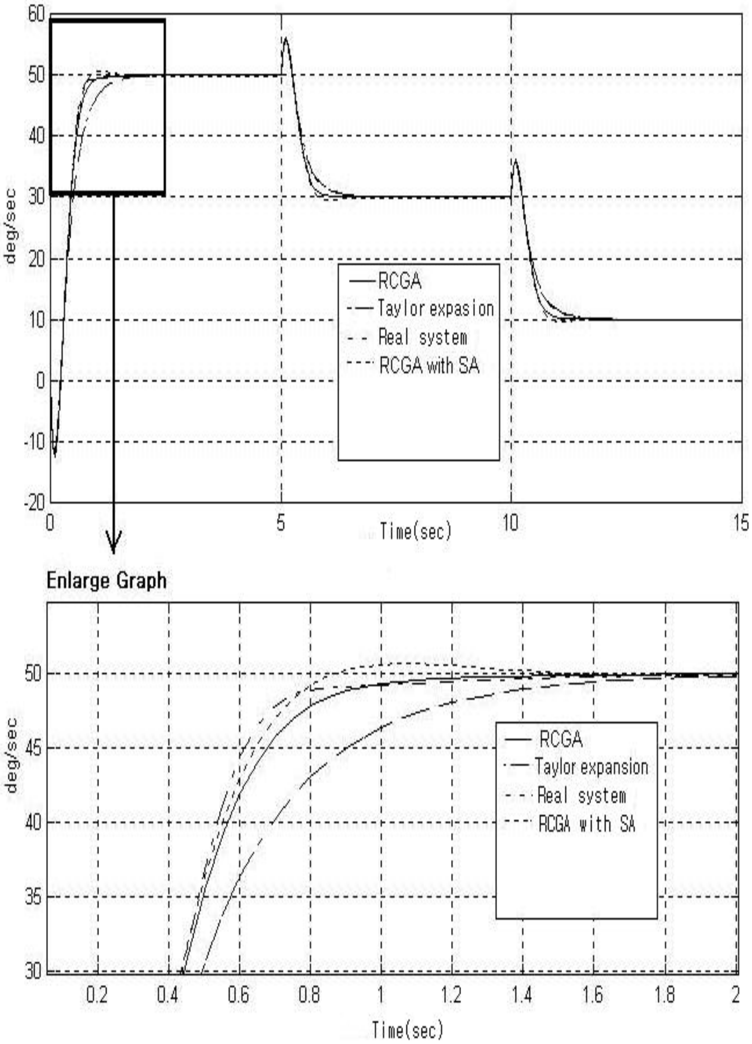


그림 4.3 **제어입력에 대한 응답**

Fig. 4.3 Responses of control input u

제 5 장 결 론

본 논문에서는 차륜형 도립진자 이동로봇의 자세 및 속도제어를 위한 최적의 선형 모델을 도출함에 있어서 SA를 접목한 RCGA알고리즘을 제안하였다. 또한 제안된 방법의 유효성을 검증하기 위하여 RCGA, Taylor 급수 전개등에 의한 방법으로 얻어진 선형모델과 동일한 입력을 인가하여 그 응답을 실제시스템의 응답과 비교 분석하였다. 그 결과 다음과 같은 결론을 도출할 수 있었다.

1) 기존의 테일러 급수 전개 또는 RCGA에 의해 얻어진 선형모델에 비해 실제 시스템의 동특성과 유사한 선형 모델의 도출이 가능했다.
2) RCGA에 SA를 접목함으로서 탐색구간을 확장함으로서 기존의 RCGA의 경우보다 더 적은 세대수에 최적해에 도달할 수 있음을 확인하였다.
3) 그러나 최적해 도달까지의 시간이 기존의 RCGA의 경우보다 길어짐을 알 수 있었다.

이러한 결과로부터 제안된 방법은 상태관측기등의 설계를 위해 보다 정도 높은 선형모델이 요구되는 경우나 점성마찰등과 같이 쉽게 측정되어질 수 없는 변수를 포함하고 있을 때 그 응용이 기대된다.

참고 문헌

[1] R.H. Canon and J.F. Shaefer, On the Control of Unstable Mechanical Systems, I.F.A.C, London, England, 1966.

[2] K. Furuta, T. Okutani and H. Sone, "Computer Control of a Double Inverted Pendulum," Comput. & Elect.Eng., Vol. 5, pp. 67-84, Nov. 1978.

[3] K. Furuta, T. Ochiai, N. One, "Attitude Control of a Triple Inverted Pendulum," Int. J. Control, Vol. 39, No. 6, pp. 1351-1365, 1984.

[4] Quing Feng and Kazuo Yamafuji, "Design and Simulation of Control Systems of an Inverted Pendulum," Robotica, Vol. 6, pp. 235-241, Mar. 1987.

[5] Yun-Su Ha and Shin'ichi Yuta, "Trajectory Tracking Control for Navigation of Self-Contained Mobile Inverse Pendulum," Proc. of IEEE/RJS International Conference on Intelligent Robots and Systems, pp. 1875-1882, 1994.

[6] 河潤秀, "平行二輪車の速度制御に關する研究," 第1回山彦シンポジウム, pp. 41-44, 1992.

[7] 주상래, 진강규, "제약조건을 가지는 컨테이너 크레인 시스템용 최적상태 궤환 제어기 설계," 한국박용기관학회지, Vol. 24, No. 2, pp. 200-206, 2000.

[8] 진강규, 유전알고리즘과 그 응용, 교우사 , 2000.

[9] 김여근, 윤복식 , 이상복, 메타휴리스틱, 영지문화사, 1999.

[10] Dan Adler, "Genetic Algorithms and Simulated Annealing: A Marriage Proposal," IEEE International Conference on Neural Network, 1993.

[11] J.H. Halland, Adaptation in natural and artificial systems, The University of Michigan Press, Michigan, 1975.

[12] D.P. Kwok and F. Sheng, "Genetic algorithm and simulated annealing for Optimal Robot Arm PID Control," Proceedings of the first IEEE Conference on Evolutionary Computation, pp. 708-713, 1994.

[13] 江口　純司, "組合せ最適化問題におけるシミュレーテッドアニーリング法による解の探索のダイナミクスに関する研究," 第3回山彦シンポジウム, pp. 101-109, 1998.

[14] Morten Irgens, "Why Simulated Annealing Works and Why it Doesn't," http://fas.sfu.ca/~mirgens.

[15] Bruce E. Rosen, 中野 良平, "シミュレーテッドアニーリング -基礎と最新技術-," 人工知能學會, Vol. 9, No. 3, pp. 365-372, May. 1995.

[16] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, "Optimization by Simulated Annealing," in Science, Vol. 220, pp.671-680, May. 1983.