[UCI]I804:24011-200000233932

2007年度 2月
博士學位論文

교류서보전동기의 속도제어를 위한 GA-PI 제어기 설계

朝鮮大學校 大學院

電 子 工 學 科

金 奐

2007年度 2月 博士學位論文 교류서보전동기의 속도제어를 위한 GA-PI 제어기 설계 金 奐

2007年度 2月
博士學位論文

교류서보전동기의 속도제어를 위한 GA-PI 제어기 설계

2007年 2月 日

朝鮮大學校 大學院

電 子 工 學 科

金 奐

교류서보전동기의 속도제어를 위한 GA-PI 제어기 설계

Design of a GA-PI Controller for Speed Control of AC Servo Motors

2007年 2月 日

朝鮮大學校 大學院

電 子 工 學 科

金 奐

교류서보전동기의 속도제어를 위한 GA-PI 제어기 설계

指導敎授　朴世承

이 論文을 工學博士 學位 論文으로 提出함.

2006年　10月　　日

朝鮮大學校 大學院

電 子 工 學 科

金　奐

金　奐의 博士學位 論文을 認准함.

委員長	朝鮮大學校	朴 光 埰	印
委　員	仁川大學校	崔 鎭 卓	印
委　員	朝鮮大學校	朴 鍾 安	印
委　員	朝鮮大學校	韓 承 朝	印
委　員	朝鮮大學校	朴 世 承	印

2006年　12月

朝鮮大學校 大學院

목 차

ABSTRACT

Ⅰ. 서 론 ····· 1

Ⅱ. 교류서보전동기의 구동특성 ····· 3

2.1. 교류서보전동기의 수학적 모델링 ····· 6

2.2. 교류서보전동기의 토크 발생원리 ····· 12

2.3. 벡터제어 ····· 15

Ⅲ. PI제어 ····· 19

3.1. PI제어기 ····· 19

3.2. PI제어의 시간영역 해석과 설계 ····· 23

3.3. PI제어의 주파수 영역 해석과 설계 ····· 24

Ⅳ. 유전자알고리즘 ····· 29

4.1. 유전자알고리즘의 진화 ····· 29

4.2. 유전자 알고리즘의 연산자 ····· 33

4.3 유전자 알고리즘의 적합도 평가 ····· 38

4.4 유전자알고리즘의 특성 ····· 41

4.5 유전자알고리즘의 적용 ····· 42

4.6 유전자알고리즘 흐름도 ····· 49

Ⅴ. 유전자 알고리즘 PI제어기 설계 ····· 51

5.1 유전자 알고리즘 PI 제어기 ····· 51

5.2 교류서보전동기의 모델링과 속도제어 ····· 53

Ⅵ. 시뮬레이션 및 실험 ····· 57

6.1 시뮬레이션 ····· 57

6.2 실험장치의 구성 ····· 65

6.3 실험결과 고찰 ····· 70

Ⅶ. 결 론 ····· 80

참 고 문 헌

List of table

Table 4.1 Compare the language of the nature with genetic algorithm ····· 42

Table 6.1 Genetic algorithm parameters ····· 57

Table 6.2 Parameters of 11[kW]servo motor ····· 59

List of figure

Fig. 2.1 Schematic diagram of three-phase synchronous motor 7

Fig. 2.2 Section diagram of two poles servo motor ····· 12

Fig. 2.3 Principle diagram of torque generation ····· 14

Fig. 2.4 Vector diagram of AC servo motor ····· 15

Fig. 2.5 Voltage vector diagram of stationary state ····· 16

Fig. 2.6 Vector locus of AC servo motor T_{max} ····· 17

Fig. 2.7 Characteristics of field control ····· 17

Fig. 3.1 PI Controller system ····· 19

Fig. 3.2 Implementation of op amplification about PI controller

$G_c(s)=K_p+\frac{Ki}{s}$ ····· 21

Fig. 3.3 Pole-zero placement of PI controller ····· 24

Fig. 3.4 Bode diagram of PI controller $G_c(s)=K_p+\frac{Ki}{s}$ ····· 25

Fig. 4.1 Flow chart of genetic algorithm ····· 31

Fig. 4.2 Crossover ····· 35

Fig. 5.1 Block diagram of control system using genetic algorithm

controller for AC servo motor ····· 51

Fig. 5.2 Block diagram of the proposed control system for AC

servo motor ····· 54

Fig. 6.1 Output characteristics of P controller ····· 58

Fig. 6.2 Changes of fitness by evolution ····· 58
Fig. 6.3 Output characteristics of PI controller ····· 58
Fig. 6.4 Speed characteristics with PI controller using genetic algorithm ····· 60
Fig. 6.5 Speed characteristics with PI controller ····· 60
Fig. 6.6 Low-speed characteristics with PI ····· 62
Fig. 6.7 Low-speed characteristics with PI controller using genetic algorithm ····· 62
Fig. 6.8 Starting characteristics by PI controller ····· 63
Fig. 6.9 Starting characteristics by PI controller by genetic algorithm ····· 63
Fig. 6.10 Speed characteristics with PI controller using genetic algorithm ····· 64
Fig. 6.11 Driving circuit of servo system ····· 65
Fig. 6.12 System of AC servo motor ····· 66
Fig. 6.13 Flow chart of the control software ····· 69
Fig. 6.14 Speed response characteristics using PI controller ····· 74
Fig. 6.15 Speed response characteristics of PI controller using genetic algorithm ····· 79

Nomenclature

V_{sa} , V_{sb} , V_{sc}	: 고정자 a, b, c상에 인가되는 전압의 순시치[V]
V_{ra} , V_{rb} , V_{rc}	: 회전자 A, B, C상에 인가되는 전압의 순시치[V]
i_{sa} , i_{sb} , i_{sc}	: 고정자 a, b, c상에 인가되는 전류의 순시치[V]
i_{ra} , i_{rb} , i_{rc}	: 회전자 A, B, C상에 인가되는 전류의 순시치[V]
λ_{sa} , λ_{sb} , λ_{sc}	: 고정자 a, b, c상의 자속 [wb]
λ_{ra} , λ_{rb} , λ_{rc}	: 회전자 A, B, C상에 인가되는 전류의 순시치[V]
L_{saa}, L_{sbb} , L_{scc}	: 고정자 a, b, c상의 자기 인덕턴스 [mH]
L_{raa}, L_{rbb} , L_{rcc}	: 회전자 A, B, C상의 자기 인덕턴스 [mH]
L_{rab}, L_{rbc} , L_{rca}	: 회전자 상간의 인덕턴스 [mH]
L_{sraa}, L_{srbb} , L_{srcc}	: 고정자와 회전자간의 상호 인덕턴스 [mH]
V_{sd} , V_{sq}	: d, q 축의 고정자 순시전압 [V]
V_{rd} , V_{rq}	: d, q 축의 회전자 순시전압 [V]
i_{sd} , i_{sq}	: d, q 축의 고정자 순시전류 [A]
i_{rd} , i_{rq}	: d, q 축의 회전자 순시전류 [A]
λ_{sd} , λ_{sq}	: d, q 축의 고정자 자속 [wb]
λ_{rd} , λ_{rq}	: d, q 축의 회전자 자속 [wb]
r_s	: 고정자 저항 [Ω]
r_r	: 회전자 저항 [Ω]
ω_s	: 고정자에 인가되는 전압 · 전류의 가속도 [rad/sec^2]
ω_r	: 회전자의 각속도 [rad/sec]
S	: 슬립(slip)
Θ_{rd}	: 회전자 d축에서 상회전의 각도 [°]
Θ_s	: 고정자에 대응하는 상에서 회전자의 각도 [°]

T	: 전기토크 [Nm]
J	: 관성계수[kgm^2]
B'	: 점성 마찰 계수[Nm/rad/sec]
V_m	: 고정자 전압의 최대치 [V]
α	: 고정자 전압의 위상 [rad]
P	: 미분 연산자 $(\frac{d}{dt})$
P_r	: 레졸버의 펄스 수
P_c	: 클럭 펄스 수
K_p	: 비례이득 정수
K_i	: 적분이득 정수
K_d	: 미분이득 정수

ABSTRACT

Design of a GA-PI Controller for Speed Control of AC Servo Motors

Kim, Hwan
Advisor : Prof Park, Sei-Seung Ph.D
Department of Electronic Engineering,
Graduate School of Chosun University

Instead of using general controllers, genetic algorithm based on a PI is used to control AC servo motors. Genetic algorithm techniques are considered as tools of artificial intelligence and are finding widespread applications in different areas of science and engineering problems. AC servo systems are competed with DC servo systems for motion control. Because of their favorable electrical, mechanical properties, good dynamics and a high efficiency, the AC servo motor has covered most servo-applications in robotics, machine tools and positioning devices.

Permanent magnet synchronous motor is receiving attention for servo drive applications during recent years because of its high torque to inertia ratio, superior power density and high efficiency. Recent development of high energy materials such as neodymium has focussed attention on wide variety of speed and position control applications.

The PI controller has the features of simple structure,

stability and reliability. But it has limitations in complex system control and can not remain above virtues under the conditions of parameters and environment uncertainties. With development of microprocessor and power electronics, numerous adaptive control, optimal control, robot control and so on, have been practically implemented in many control schemes. Comparing with PI, these methods have nonlinear control and highly complex system but they depend on accurate model excessively and need long time and complex computation. The effects of modern control theory are diminishe when unmodeled dynamics of the system emerge.

In order to reduce mechanical parameter variation, and to obtain a good dynamic performance at the viewpoint and wide speed range, an intelligent controller is employed. The genetic algorithm can be adopted widely, however the convergence speed is limited by its constant learning rate. The genetic algorithm based self tuner of coefficient K_p, K_i consists of position value.

In this paper, genetic algorithm based on a PI is proposed and executed in a DSP, for the implementation of high performance speed control system. A 32-bit DSP, TMS320LF2407 has been used, which is capable of high speed and floating point calculation. The design technicue of self tuning PI controller using genetic algorithm with the newly proposed learning algorithm is presented, and the designed controller is used to the accurate position control of unmodeled AC servo systems with uncertainty. To show the effectiveness of the proposed scheme, several simulation studies has been performed. Computer simulations were performed to show the usefulness of the proposed controller in case of AC servo motor control. As shown in results of

simulations, proposed controller has good performance and robust properties even though disturbance exists.
The genetic algorithm has been used to find quickly, random practical optimum solution, can solve problems concentrating to local minima because of taking many input at the same time as well as an input of reference space.
Therefore, there is no continuously differential condition in the object function, is able to parallel calculation and same time-optimization as encoding many variable to one string. We executed controller to apply driver of AC Servo motors with genetic algorithm-PI controller(GA-PI Controller) to realize precision control and application of robot actuator in industry.

The proposed controller for good tracking command values for motor driving is verified through the computer simulation. The system is designed to improve set-point tracking capability. The simulation and experimental results show that this genetic algorithm system provides precise response.

Ⅰ. 서 론

최근 급속한 산업화로 인해 단순 반복 작업 등의 일 등은 기계가 대신하게 되었다. 이러한 산업용 기기에 사용되는 전동기는 적용범위의 확대, 고속 · 정밀화의 목적에 따라 종류도 다양하고, 현대사회에 널리 중요하게 쓰임에 따라, 이러한 중요한 역할에 교류서보전동기가 널리 이용이 되고 있다. 교류서보전동기는 전자부품 기술의 발전과 각종 기계기술의 진보와 직류서보전동기의 단점으로 지적되어 온 기계적 정류기구를 제거시켰을 뿐만 아니라 신뢰성과 내환경성이 크게 향상되었다. 하지만 직류서보전동기는 정류자를 가지고 있어서 정류한계에 의해 고속토크가 높은 상태에서 사용이 극히 제한되고, 브러시가 시간이 지난 후 마모되는 단점으로 인해 점검과 교체가 필요하다. 이러한 문제점을 해결하기 위하여 스스로 적응할 수 있는 제어기에 관한 연구가 활발하다.[1]

많은 종류의 제어기 가운데 PI 제어기는 그 구조가 간단하고 견실하며 특성이 좋기 때문에 산업계에 널리 이용되고 있다. PI 제어기는 원하는 특성을 얻기 위하여 각 계수들을 적절하게 결정하여야 하는데 이에 대한 방법으로는 최적화 방법, 시행착오법 등이 있다. 이러한 방법들은 시스템 매개변수를 잘 알아야 하며 동작 중에 시스템의 매개변수가 변화하면 PI 계수들을 다시 조정해주어야 한다. 따라서 제어 시스템의 구조가 복잡하거나 정확히 알 수 없는 경우에는 대상 시스템의 특성을 정확하게 나타내는 수학적 모델을 구하기가 어려울 뿐만 아니라 계수들을 적절하게 조정하기가 어렵다.[2]

인공유전 시스템인 유전자 알고리즘은 1975년 John Holland에 의해 개발된 전역적인 최적화 알고리즘으로 자연세계의 진화현상에 기초한 계산 모델이다. 유전자 알고리즘은 지금까지의 방법으로는 해결이 곤란한 문제에 대해서 실용상의 최적해를 랜덤하고 빠르게 찾는 방법으로써 사용되어 왔다.[3]

또한, 유전자 알고리즘은 기존의 최적화 이론들과는 근본적으로는 다른 중요한 특징을 가지고 있다. 즉 목적함수에 연속미분조건이 없으며 다수의 변수들을 하나의 문자열로 부호화 함 으로써 병렬연산 및 동시최적화가 가능하다. 이러한 이유들로 인해 유전알고리즘은 최근 주목받는 최적화 기법중의 하나가 되었으며 특히 공학적인 분야에서 많은 적용사례들이 발표되고 있다. 유전자 알고리즘은 자연선택의 원리와 자연계의 생물 유전학에 기본이론을 두며 병렬적이고 전역적인 탐색알고리즘으로서 모든 생물은 주어진 환경 속에 적응함으로써 살아남는다는 다윈의 적자생존의 이론을 기본개념으로 한다. 유전자알고리즘은 해결하고자 하는 가능한 해들을 정해진 형태의 자료구조로 표현한 다음, 이를 변형하여 점점 더 좋은 해로 만든다.[4~5]

본 논문에서는 유전자알고리즘을 이용한 교류서보전동기의 속도제어 방법을 제안하였다. 유전자알고리즘은 최근 최적화 탐색 문제에서 점차 중요한 알고리즘으로 자리잡혀가고 있다. 이것은 다른 최적화기법에 비하여 먼저 파라미터 영역의 연속성이나 미분가능과 같은 수학적 조건이 필요하지 않고, 국부탐색이 아닌 전역 탐색을 하므로 국부수렴의 확률이 적으며, 알고리즘 수행을 위한 보조정보가 필요하지 않다. 유전자알고리즘은 전형적인 최적의 해를 발견할 높은 가능성을 가진다는 장점과 목적 함수값에 대한 수학적 제약이 거의 없기 때문에 위 이득 값 등의 최적 해를 구할 수 있다.

따라서 본 논문에서는 목적함수에 연속미분조건이 없으며 다수의 변수들을 하나의 문자열로 부호화 함 으로써 병렬연산 및 동시최적화가 가능한 교류서보전동기의 제어기를 DSP인 TMS320LF2407로 설계제작 하여 기존의 PI제어기와 유전자알고리즘을 이용한 PI제어기에 대한 시뮬레이션과 실험결과를 통해 제안한 알고리즘의 우수성을 입증하고자 한다.

Ⅱ. 교류서보전동기의 구동특성

로봇의 관절 등을 움직여 주는 구동부에는 AC서보 전동기, DC 서보전동기 등의 각종 전동기가 사용될 수 있으나 스텝모터는 제어가 용이하고 오차가 누적되지 않으며 디지털제어에 적합하다는 등의 장점이 있는 반면 유연성이 부족하고 고속운전이 어려우며 진동에 의하여 원활한 운동이 불가능할 뿐만 아니라 외란에 의한 오차가 발생할 경우에 대비하여 부수적인 회로가 필요하다.[6~8]

교류서보전동기는 전력전자의 발전과 고성능 영구자석의 개발과 양산화에 의해 직류서보전동기의 제어상의 장점을 갖으면서 기계적 정류기구를 제거시켰을 뿐만 아니라 신뢰성과 내환경성이 크게 향상되었다. 교류서보전동기에는 IM형과 SM형이 있는데 SM형인 영구자석 동기전동기(PMM)는 회전자가 영구자석으로 구성되어 있어 관성 모멘트가 적고 회전수에 동손이 적어 효율이 높고 가속성능이 양호하여 직류기에서 갖는 여러 가지 제약성을 갖고 있지 않아서 자동차 공업 생산라인에서의 로봇, 반도체 기기나 우주산업분야, 원자력 관련 기기나 장치 등 특수환경 등에 널리 사용되고 있다.[9~11]

지난 30년 동안 진화 및 유전원리에 기초한 문제해결 시스템에 대한 관심이 증가하고 있는데, 이러한 시스템들은 해가 될 가능성이 있는 개체집단을 유지하고, 개체의 적합도와 몇 가지 “유전” 연산자에 기초한 선택과정을 가지고 있다. 이들 시스템의 한 형태는 진화전략(Evolution Strategies)의 부류로, 매개변수 최적화문제에 대하여 자연의 진화원리를 흉내 낸 알고리즘이다.[12] 또 다른 형태의 진화기반 시스템은 Holland의 유전자 알고리즘이다. 1990년에 Koza는 진화기반 시스템인 유전자 프로그래밍을 제안하여 어떤 문제를 해결하기 위한 가장 적합한 컴퓨터 프로그램을 찾아내었다. 즉, 외부환경에 맞도록 자기 자신의 내부구조를 변화시켜 조절하는 학습능력을 가지고 있다. 유전자 알고리즘에 기반을 둔 많

은 진화 프로그램들은 주어진 문제에 대해 많은 진화 프로그램들이 체계화될 수 있다. 이들 프로그램은 여러 가지 면에서 다를 수 있는데, 어떤 개체와 그것을 변화하는 “유전” 연산자, 초기 개체집단을 생성하는 방법, 문제의 구속조건을 취급하는 방법, 그리고 매개변수들(개체집단의 크기, 여러 연산자들의 적용확률 등)을 구현하기 위하여 다른 데이터 구조를 사용할 수 있다. 그러나 개체집단은 어떤 변화과정을 겪으며, 진화과정 동안 개체는 생존을 위하여 투쟁한다는 공통원리를 가지고 있다.[13~14]

진화 프로그래밍의 아이디어는 새로운 것이 아니고 적어도 30년 동안이나 우리 주위에 존재하여 왔다. 그러나 진화프로그램에 대한 개념은 유전자 알고리즘의 아이디어에 전적으로 기초하고 있다.

교류서보전동기는 동기기형과 유도기형으로 대별되며 동기형인 영구자석 동기전동기는 회전자가 영구자석으로 구성되어 있어 관성모멘트가 적고 회전수에 동손이 적어 효율이 높고 가속성능이 양호하다. 직류기에서 갖는 여러 가지 제약성을 갖고 있지 않아서 자동차 공업 생산라인에서의 로봇, 반도체 기기나 우주산업분야, 원자력 관련 기기나 장치 등 특수 환경 등에 널리 사용되고 있다.[15]

교류서보전동기는 정류자를 대신하는 자극위치검출기구와 정현파 3상 교류 제어 기구를 필요로 하는데 고속, 고 토크 영역까지 사용이 가능하므로 설치 환경에의 적응성이 높은 이점이 있다. 브러시가 없어서 보수가 용이하고 내환경성이 우수하며 정류한계가 없어서 고속으로 고 토크 이용이 가능하고 코일이 고정자에 있어서 방열상 유리하나 자극위치검출이 필요하고 제어기가 복잡해서 고가로 취급된다. 따라서 교류서보전동기에서는 자극위치검출을 하고 있고 자석의 종류와 자극구조에 따라서 회전자를 고정자에서 빼면 감자가 일어난다.[16]

동기기형 교류서보전동기는 전기자 전류와 토크관계가 선형이고 제어가 용이하며 비상 정지 시는 다이나믹 브레이크의 사용이 가능하다. 그러나 모우터에 영구자석을 고정시키기 위해서 구조가 다소 복잡해지고 제어를

위해서 로우터 위상을 검출할 필요가 있다. 회전자에 영구자석을 끼워 넣고 고정자에 전기자 코일을 설치한 회전계자형 동기기로서 기본원리는 회전자와 고정자 사이에 토크를 발생시키는 방식이고 전기자의 3상전류를 자극과 직교하도록 제어하는 것이다. 서보전동기로서 필요로 하는 소형경량, 고빈도 가·감속, 대출력과 보수·유지비의 감소를 충족시키기 위해서 회전자에는 고성능 자속으로서 희토류와 페라이트, 알니코 등을 사용한다.

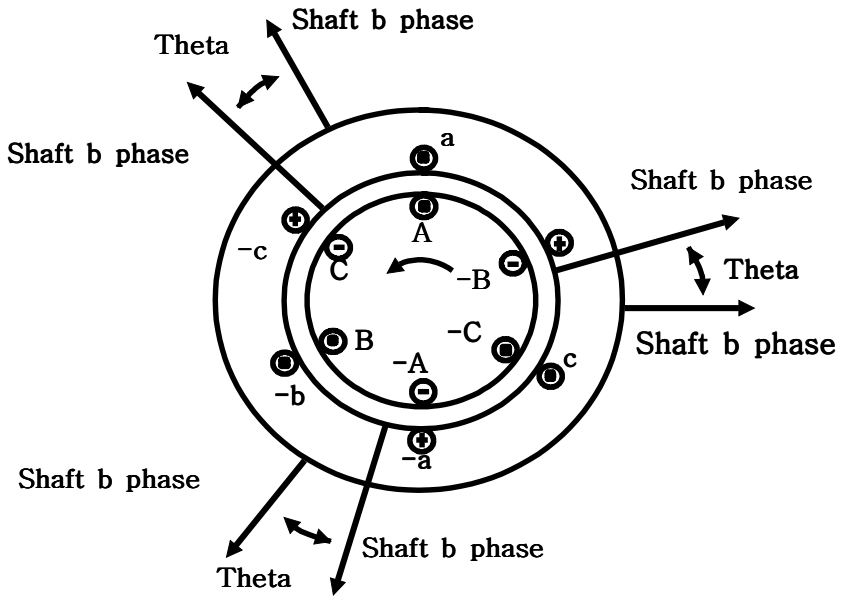
유도기형 교류서보전동기는 구조가 간단하고 튼튼한 상자형 모터를 사용하는 교류서보 전동기방식으로써 고속 대출력용으로 적합하다. 유도기형은 동기기형에 비하면 형태도 크고 공극을 아주 작게 하지 않는 한 역률, 효율에서 떨어지고 소 용량기의 분야에는 적용되지 않는다. 회전자의 구조가 간단하고 검출기도 특수한 것을 필요로 하지 않고 직류서보전동기의 것과 동일한 것을 사용할 수 있다. 그러나 정지 시에도 상시 여자전류를 흘려 둘 필요가 있기 때문에 그것에 의한 발열의 문제가 있고 비상정지의 경우에 교류서보전동기와 같이 전기자 권선을 단락하여 다이나믹 브레이크를 걸 수 없다. 유도기형 교류서보전동기의 제어방식에는 전동기에 자속검출 센서를 채택함으로서 정밀도 높은 제어를 할 수 있는 자속검출형 벡터제어방식과 센서를 사용하지 않았기 때문에 전동기 구조는 간단하지만 자속의 예측제어기능을 갖지 않으면 안 되고 제어장치 자체가 약간 복잡하게 되는 미끄럼 주파수형 벡터 제어방식이 있다. 전동기의 출력토크는 계자 자속분과 토크 전류분의 곱에 비례한다. 전동기의 출력 토크는 계자 자속분과 토크 전류분의 곱에 비례한다. 유도기형에 있어서는 이 양 성분을 합성한 전기량으로써 제어하는 것이 필요하고 서보드라이버도 벡터 제어방식으로 된다.[17]

교류서보전동기는 그 용량범위가 넓고 또 높은 신뢰성과 보수, 유지 면에 이점이 있으므로 적용에 대해서는 그 범위가 넓고 고정자의 코어를 공냉하고 있기 때문에 냉각능력이 크고 회전자의 구조에 독특한 방식을 채

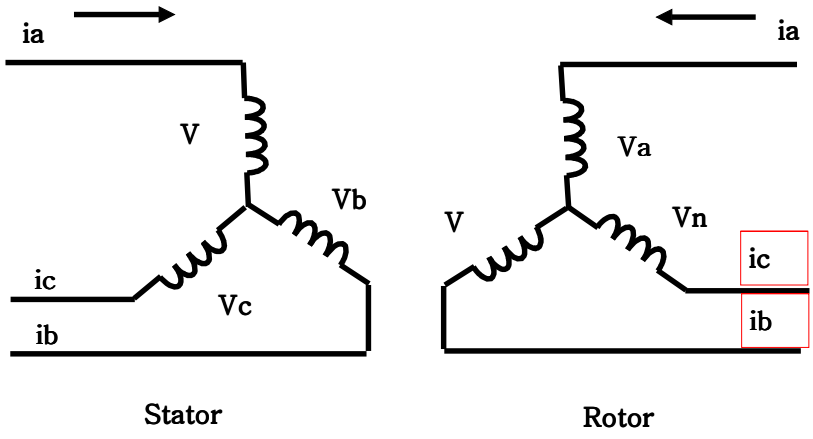
용하여 소형 콤팩트로서 큰 토크/이너셔비를 얻고 있으며 직류서보전동기와 같이 정류 불꽃에 의하여 최고속도가 제한되는 일이 없기 때문에 고속영역에 있어서의 토크 특성이 우수하다.[18]

2.1 교류서보전동기의 수학적 모델링

그림 2-1은 동기전동기의 구조로써 (a)는 3상 동기전동기의 고정자와 회전자의 자기적인 결합회로를 나타내고 (b)는 등가회로를 나타내고 있다.



(a) Magnetically stator and rotor circuit



(b) Equivalent Circuit

Fig. 2.1 Schematic diagram of three-phase synchronous motor

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = r_1 \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + P \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{2-1}$$

회전자 권선의 전압방정식은 아래 식 (2-2)와 같다.

$$\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = r_2 \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + P \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{2-2}$$

또한, 고정자 측에서 상 자속은 식(2-3)과 같고,

$$\begin{bmatrix}\lambda_A\\ \lambda_B\\ \lambda_C\end{bmatrix}=\begin{bmatrix}L_{aa} & L_{ab} & L_{ab}\\ L_{ab} & L_{aa} & L_{ab}\\ L_{ab} & L_{ab} & L_{aa}\end{bmatrix}\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix}+L_{aA}$$

$$\begin{bmatrix}\cos\Theta_2 & \cos(\Theta_2-\frac{4}{3}\pi) & \cos(\Theta_2-\frac{2}{3}\pi)\\ \cos(\Theta_2-\frac{2}{3}\pi) & \cos\Theta_2 & \cos(\Theta_2-\frac{4}{3}\pi)\\ \cos(\Theta_2-\frac{4}{3}\pi) & \cos(\Theta_2-\frac{2}{3}\pi) & \cos\Theta_2\end{bmatrix}\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix} \qquad (2\text{-}3)$$

회전자 측에서 상 자속은 아래 식(2-4)와 같다.

$$\begin{bmatrix}V_A\\ V_B\\ V_C\end{bmatrix}=L_{aA}\begin{bmatrix}\cos\Theta_2 & \cos(\Theta_2-\frac{4}{3}\pi) & \cos(\Theta_2-\frac{4}{3}\pi)\\ \cos(\Theta_2-\frac{2}{3}\pi) & \cos\Theta_2 & \cos(\Theta_2-\frac{2}{3}\pi)\\ \cos(\Theta_2-\frac{4}{3}\pi) & \cos(\Theta_2-\frac{4}{3}\pi) & \cos\Theta_2\end{bmatrix}\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix}$$

$$+\begin{bmatrix}L_{AA} & L_{AB} & L_{AB}\\ L_{AB} & L_{AA} & L_{AB}\\ L_{AB} & L_{AB} & L_{AA}\end{bmatrix}\begin{bmatrix}i_A\\ i_B\\ i_C\end{bmatrix} \qquad (2\text{-}4)$$

고정자 전류, 자속, 전압을 d-q변환하면 식(2-5), (2-6), (2-7)과 같고 q축은 d축보다 $\frac{\pi}{2}$만큼 앞선다.

$$\begin{bmatrix}i_{1d}\\ i_{1q}\\ i_{10}\end{bmatrix}=[A]\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix} \qquad (2\text{-}5)$$

$$\begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{10} \end{bmatrix} = [A] \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{2-6}$$

$$\begin{bmatrix} V_{1d} \\ V_{1q} \\ V_{10} \end{bmatrix} = [A] \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{2-7}$$

$$[A] = \frac{2}{3} \begin{bmatrix} \cos\omega t & \cos(\omega t - \frac{2}{3}\pi) & \cos(\omega t - \frac{4}{3}\pi) \\ -\sin\omega t & \sin(\omega t - \frac{2}{3}\pi) & -\sin(\omega t - \frac{4}{3}\pi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2-8}$$

여기서 [A]는 식(2-9)과 같고, 변 i_{10}, λ_{10}, V_{10} 는 d, q 변환에 필요한 것으로 평형 3상 전동기에서는 0 이다.

회전자 권선에 대해여 d, q 변환하면 아래 식과 같고, d, q 축은 ω_2와 동기 하여 회전하고 회전자에는 아래 식과 같은 슬립 S가 발생한다.

$$\frac{d\Theta_s}{dt} = P\Theta_s = S_\omega \tag{2-9}$$

$$\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix} = [A] \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \tag{2-10}$$

$$\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix} = [A] \begin{bmatrix} \lambda_A \\ \lambda_B \\ \lambda_C \end{bmatrix} \tag{2-11}$$

$$\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix} = [A] \begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} \tag{2-12}$$

$$[A]=\frac{2}{3}\begin{bmatrix} \cos\Theta_s & \cos(\Theta_s-\frac{2}{3}\pi) & \cos(\Theta_s-\frac{4}{3}\pi) \\ -\sin\Theta_s & \sin(\Theta_s-\frac{2}{3}\pi) & -\sin(\Theta_s-\frac{4}{3}\pi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad (2\text{-}13)$$

여기서 [A′]은 위 식과 같고 변수 i_{20}, λ_{20}, V_{20} 는 d, q 변환에 필요한 것이며 평형 3상 전동기에서는 0 이다. $\Theta_2=\omega t-\Theta_s$의 관계를 이용하여 자석방정식을 간략히 하면 아래식과 같다.[19]

$$\begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{10} \end{bmatrix}=L_{11}\begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{10} \end{bmatrix}+L_{12}\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix} \qquad (2\text{-}14)$$

$$\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix}=L_{22}\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix}+L_{12}\begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{20} \end{bmatrix} \qquad (2\text{-}15)$$

식(2-1), (2-5), (2-6), (2-7)로부터 다음 식이 성립하고

$$\begin{bmatrix} V_{1d} \\ V_{1q} \\ V_{10} \end{bmatrix}=r_1\begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{10} \end{bmatrix}+[A][A]^{-1}\begin{bmatrix} \lambda_{1d} \\ \lambda_{1q} \\ \lambda_{10} \end{bmatrix}+P\begin{bmatrix} \lambda_{1d} \\ \lambda_{1q} \\ \lambda_{10} \end{bmatrix} \qquad (2\text{-}16)$$

식 (2-2), (2-10), (2-11), (2-12)로부터 다음 식이 성립한다.

$$\begin{bmatrix} V_{2d} \\ V_{2q} \\ V_{20} \end{bmatrix}=r_2\begin{bmatrix} i_{2d} \\ i_{2q} \\ i_{20} \end{bmatrix}+[A'][A']^{-1}\begin{bmatrix} \lambda_{2d} \\ \lambda_{2q} \\ \lambda_{20} \end{bmatrix}+P\begin{bmatrix} \lambda_{2d} \\ \lambda_{2q} \\ \lambda_{20} \end{bmatrix} \qquad (2\text{-}17)$$

식(2-16)과 (2-17)을 정리하면 식(2-18), (2-19)와 같다.

$$\begin{bmatrix} V_{1d} \\ V_{2q} \end{bmatrix}=r_1\begin{bmatrix} i_{1d} \\ i_{1q} \end{bmatrix}+P\begin{bmatrix} \lambda_{1d} \\ \lambda_{1q} \end{bmatrix}+\omega\begin{bmatrix} -\lambda_{1q} \\ -\lambda_{1q} \end{bmatrix} \qquad (2\text{-}18)$$

$$\begin{bmatrix} V_{2d} \\ V_{2q} \end{bmatrix} = r_2 \begin{bmatrix} i_{2d} \\ i_{2q} \end{bmatrix} + P \begin{bmatrix} \lambda_{2d} \\ \lambda_{2q} \end{bmatrix} + \omega \begin{bmatrix} -\lambda_{2q} \\ -\lambda_{2q} \end{bmatrix} \tag{2-19}$$

식(2-14)과 (2-15)를 식(2-18)과 (2-19)에 대입하면 다음 식이 성립한다.

$$\begin{bmatrix} V_{1d} \\ V_{1q} \\ V_{2d} \\ V_{2q} \end{bmatrix} = \begin{bmatrix} r_1 + PL_{11} & -\omega L_{11} & PL_{12} & -wL_{12} \\ \omega L_{11} & r_1 PL_{11} & wL_{12} & PL_{12} \\ PL_{12} & -swL_{12} & r_2 + PL_{22} & -swL_{22} \\ swL_{12} & PL_{12} & swL_{22} & r_2 PL_{22} \end{bmatrix} \begin{bmatrix} i_{1d} \\ i_{1q} \\ i_{2d} \\ i_{2q} \end{bmatrix} \tag{2-20}$$

식(2-17), (2-18), (2-19), (2-20)으로 부터 교류서보 전동기의 출력 P를 구하면 다음과 같다.

$$\begin{aligned} P &= \frac{3}{2}(V_{2d} i_{2d} + V_{2q} i_{2q}) \\ &= \frac{3}{2}\{-\lambda_{2q} i_{2d}(s\omega) + \lambda_{2d} i_{2q}(s\omega)\} \end{aligned} \tag{2-21}$$

회전자 자속은 $-s\omega 2/$극수 이므로 전동기 토크 T는 식(2-22)로 나타내며

$$\begin{aligned} T &= \frac{3}{2} \cdot \frac{P_N}{2}(\lambda_{2q} i_{2d} - \lambda_{2d} i_{2q}) \\ &\frac{3}{4} \cdot P_N \cdot L_{12}(i_{1d} i_{2q}) \end{aligned} \tag{2-22}$$

이 때 기계적 토크는 식(2-23)과 같다.

$$T_m = J\omega_2 + B\omega_2 + T_d \tag{2-23}$$

2.2 교류서보전동기의 토크 발생원리

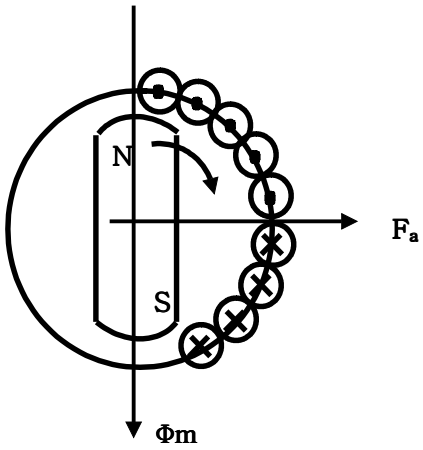


Fig. 2-2 Section diagram of two poles servo motor

그림 2-2는 2극 서보전동기의 구조를 나타낸 것으로 전기자 도체에 전류가 흐르면 이 도체는 $F = Bli[N]$ 의 힘이 발생되고 회전자에는 전기자 합성기자력 F_a 와 자속 ϕ_m 의 곱에 비례하는 토크가 발생된다.[20]

$$T \propto \phi_m \times F_a \qquad (2\text{-}24)$$

각 권선의 합성기자력 F_a 는 항상 계자 속에 수직인 관계를 이루고 있으므로 직류기의 브러시와 정류기가 전력변환기로 바뀐 것과 같으며 3상의 경우 각 전기자 권선에 흐르는 전류는 아래 식과 같다.[21]

$$I_U = I_m \sin\Theta$$

$$I_V = I_m \sin\left(\Theta - \frac{2}{3}\pi\right) \qquad (2\text{-}25)$$

$$I_W = I_m \sin\left(\Theta - \frac{4}{3}\pi\right)$$

따라서 각 상의 전기자 도체와 직교하는 공극의 자속밀도는 아래 식으로 나타내 질 수 있다.

$$B_{gU} = B_m \sin\Theta$$

$$B_{gV} = B_m \sin\left(\Theta - \frac{2}{3}\pi\right) \qquad (2\text{-}26)$$

$$B_{gW} = B_m \sin\left(\Theta - \frac{4}{3}\pi\right)$$

이 때 교류서보 전동기가 발생하는 토크는 아래 식과 같다.

$$T_e = K\left[B_{gU}\ B_{gV}\ B_{gW}\right]\begin{bmatrix} I_U \\ I_V \\ I_W \end{bmatrix} = \frac{3}{2}KB_m I_m \qquad (2\text{-}27)$$

위 식으로부터 토크는 전기자 전류와 자속밀도의 곱에 비례하며 회전자의 변위각 Θ에 무관함을 알 수 있다. 따라서 전기자 전류나 자속밀도를 변환하면 토크가 제어된다.[22]

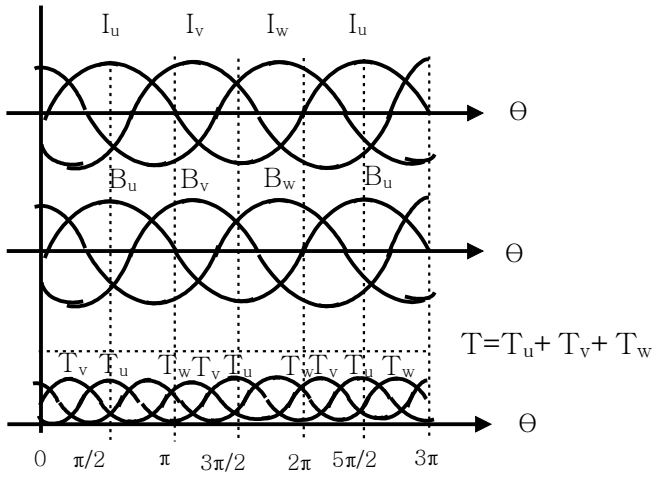


Fig. 2-3 Principle diagram of torque generation

그림 2-3은 토크발생 원리도를 나타내고 있다. 동기전동기는 슬립링을 통해서 전기자권선에 3상전류를 공급하고 회전자가 상의 순서와 반대방향으로 동기각속도를 갖고 회전한다면 전기자 전류의 분포는 고정자에 대하여 정지하고 합성기전력은 계자기전력에 대하여 임의 각도를 유지하며 정지하는 것으로 된다. 그러므로 합성기전력의 위치에 브러시축을 이동시킨 직류기라고 등가적으로 생각할 수 있으며 합성기자력이 계자기자력의 방향과 같게 되려 하는 토크를 발생하게 된다. 그러나 동기기에서 계자기자력의 위치는 전류의 위상에 따라서 결정되고 전류의 위상은 전기자에 공급되는 전압이 일정하더라도 계자기자력의 크기에 따라서 현저하게 달라진다.[23~24]

2.3 벡터제어

그림 2-4는 교류서보 전동기의 벡터도이다.

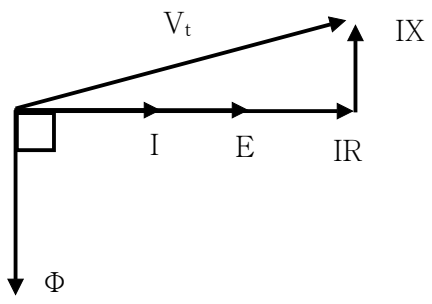


Fig. 2-4 Vector diagram of AC servo motor

동기기형 서보전동기는 계자가 영구자석이므로 직접 계자제어가 곤란하기 때문에 전기자 반작용을 이용하여 계자를 약하게 하는 계자제어법을 선택하였다. 이때의 수학적 모델링은 식(2-20)과 (2-22)로부터 다음 식을 얻을 수 있다.[25~26]

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \begin{bmatrix} R_a + PL_a & -w_m L_a \\ \omega_m L_a & R_a + PL_a \end{bmatrix} + \begin{matrix} w_m \phi \\ 0 \end{matrix}$$

$$T_\varepsilon = \frac{3}{4} \cdot (P_n) \times \phi i_d = Jw_m + Bw_m + T_L \qquad (2\text{-}28)$$

보통 $i_d = 0$ 에서 제어하므로 i_d 는 전동기 전류가 되며 정상상태에서 $PL_a i_d = PL_a i_q = 0$ 이므로 식(2-28) 로부터 다음 식이 성립한다.[27]

$$V_{ds} = R_a i_d - w_m L_a i_q + w_m \phi$$

$$V_{dq} = R_a i_q + w_m L_a i_d \qquad (2\text{-}29)$$

식(2-28)과 (2-29)로부터 $i_q = 0$ 에서 전동기 단자전압 V_o 는 순시토크에 비례하여 증가하며 I_q 를 제어하면 계자제어를 행할 수 있다.

정상상태의 전압 벡터도는 아래 그림 2-6 과 같다.[28]

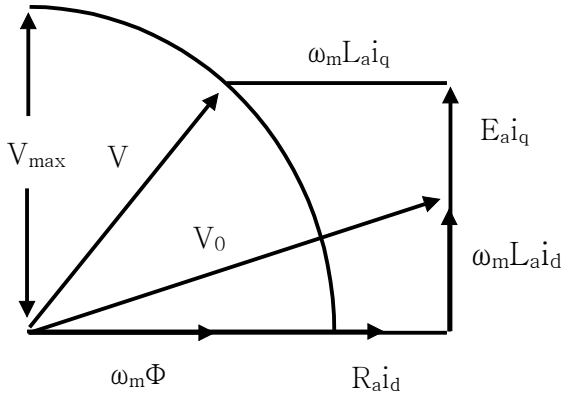


Fig. 2-5 Voltage vector diagram of stationary state

그림 2-5에서 V_o 는 $i_q = 0$ 에서 전동기 단자전압이고 i_d 를 충분히 제어하면 전동기 단자전압 V는 V_{max} 와 같게 된다.

전동기 전류 $|i| = \sqrt{(i_d)^2 + (i_q)^2}$는 허용 최대전류 I_{max} 보다 작게 제한해야 한다. 따라서 최대출력 토크도 고속 운전시에 크기가 제한되며 I_{max} 는 계자를 감자하는 요소가 된다.

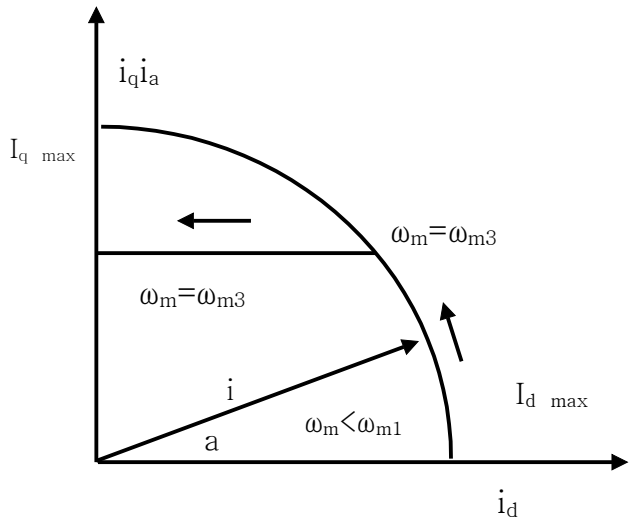


Fig. 2-6 Vector locus of AC servo motor T_{max}

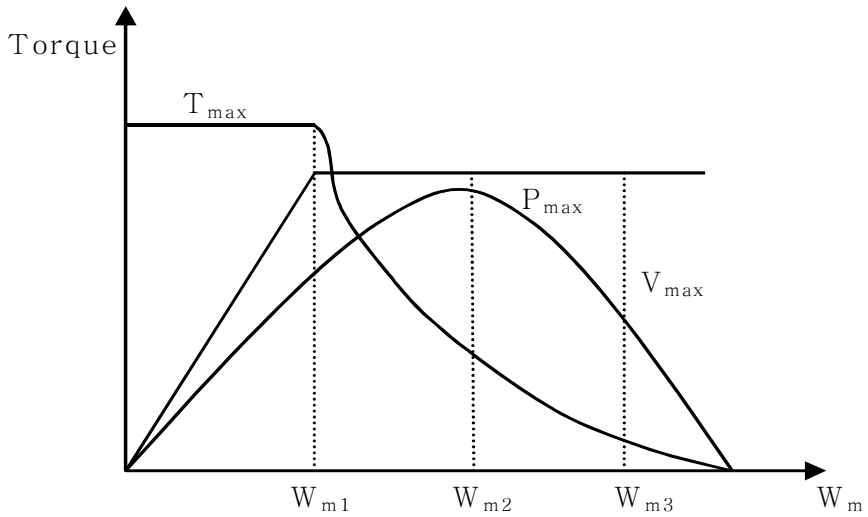


Fig. 2-7 Characteristics of field control

그림 2-6은 최대 토크 발생시의 전동기 전류의 벡터궤적을 나타내며 그림 2-7은 계자제어를 하는데 필요한 토크, 출력, 전압에 대한 전동기 속도의 관계를 나타낸다. 전동기 속도 w_m이 w_{m1}을 초과할 때 전동기 단자전압V는 i_q를 제어하며 V_{max}로 제한하고 w_m이 w_{m2}을 넣을 때 제한된다.

w_m이 w_{m3} 에 도달할 때 $i_d = T_e = 0$ 이 되며 이 때 전동기 속도 w_{m3}는 허용 최대 회전수와 레졸버 검출기 출력에서 w_m에 대하여 필요한 토크 T^*를 그림 2-6에서 구하고 식 (2-29)에서 i_d^*를 계산한다. 그리고 유기전압의 최대 허용치를 제한하였을 때 그림 2-6에서 필요한 i_d^*를 구하며 i_d^* 와 i_q^*로부터 그림 2-6 로부터 a^*를 결정한다.

$$i_d^* = i^* \cos a^*$$

$$i_q^* = i^* \sin a^* \qquad (2\text{-}30)$$

식 (2-30)을 d-q 역 변환하면 i_a^*, i_b^*, i_c^*를 지령치로서 i_a, i_b, i_c를 검출하여 피이드백 하고 그 차의 신호를 펄스변조에 사용된다. 이 계산은 마이크로프로세서로서 행하며 *는 지령치를 나타낸다.

$$i_a^* = i^* \cos(wt + a^*)$$

$$i_b^* = i^* \cos(wt - \frac{2}{3}\pi + a^*)$$

$$i_c^* = i^* \cos(wt - \frac{4}{3}\pi + a^*) \qquad (2\text{-}31)$$

Ⅲ. *PI*제어

3.1 PI 제어기

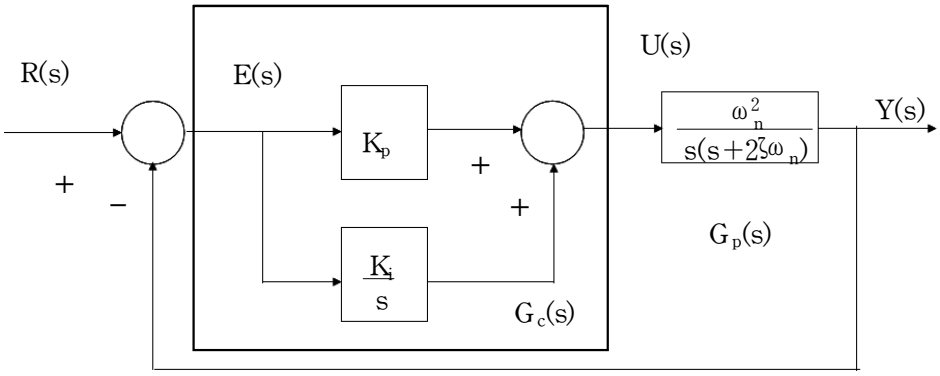


Fig 3.1 PI control system

그림 3.1은 직렬 PI 제어기를 갖는 전형적인 2차 시스템을 나타낸 것이다. PI제어기의 전달함수는

$$G_c = K_p + \frac{Ki}{s} \tag{3-1}$$

이다.[29] 여기서 K_p가 주어질 때, K_i의 값이 너무 작지 않게 되든지 op 증폭회로 구현시의 용량이 너무 크게 된다. 식(3-1)의 두 op 증폭회로에 의한 구현이 그림 3.2와 같이 된다. 그림 3.2(a)의 두 op 증폭회로의 전달함수는 다음과 같다.

$$G_c(S) = \frac{E_o(s)}{E_{in}(s)} = \frac{R_2}{R_1} + \frac{R_2}{R_1 C_2 s} \tag{3-2}$$

식 (3-1)을 식(3-2)와 비교하면,

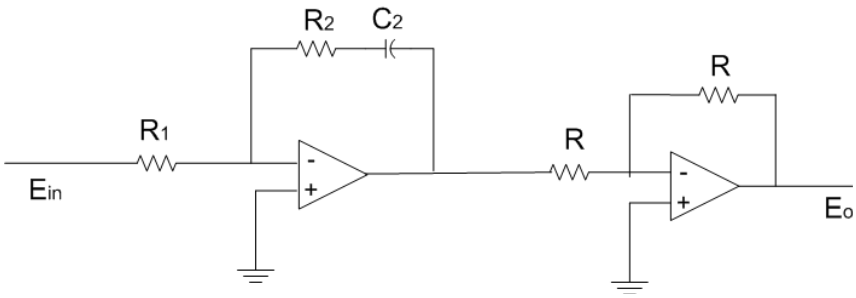
$$K_p = \frac{R_2}{R_1}, \qquad K_i = \frac{R_2}{R_1 C_2} \tag{3-3}$$

를 얻는다. 그림 3-2(b)의 세 op 증폭회로의 전달함수는

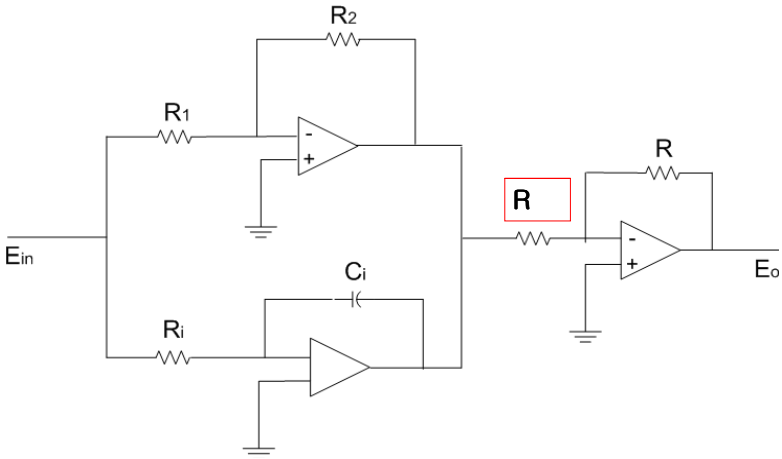
$$G_c(s) = \frac{E_o(s)}{E_{in}(s)} = \frac{R_2}{R_1} + \frac{1}{R_i C_i s} \tag{3-4}$$

이다. 따라서 PI제어기의 파라미터들은 다음과 같이 주어진다.[30]

$$K_p = \frac{R_2}{R_1}, \quad K_i = \frac{R_2}{R_i C_i} \tag{3-5}$$



(a) Two op amplification circuits



(b) Three op amplification circuits

Fig. 3.2 Implementation of two op amplification circuits about

PI controller $G_c(s)=K_p+\frac{K_i}{s}$

그림 3-2(b)의 회로를 이용할 경우의 장점은 K_p와 K_i의 값을 독립적으로 회로 파라미터에 관련시킬 수 있다는 점이다. 그러나 어느 회로에서나 K_i의 값이 작아야 되고 따라서 큰 값의 커패시터 용량에 부딪치게 된다. 보상된 시스템의 전방경로 전달함수는

$$G(s) = G_c(s)G_p(s) = \frac{\omega^2(K_p s + K_i)}{s^2(s + 2\zeta\omega_n)} \qquad (3\text{-}6)$$

이다. PI제어기의 직접적인 효과는 다음과 같다.[31]

1. $s = -K_i/K_p$에 있는 영점을 전방경로 전달함수에 추가한다.
2. $s = 0$에 있는 극을 전방경로 전달함수에 추가한다. 이는 시스템의 형태를 한 단계 증가시켜 2형태 시스템으로 변경함을 의미한다. 따라서 원래 시스템의 정상상태오차를 한 차수 개선한다. 즉 주어진 입력에 대한 정상상태오차가 상수이면 PI 제어는 이를 영으로 줄인다(보상된 시스템이 안정을 유지하는 경우).

PI제어가 시스템의 형태를 한 단계 증가시키며 따라서 1차만큼 정상상태오차를 개선한다. 그림 3.1의 시스템이 식(3-6)으로 표현된 전방경로 전달함수를 가질 때 기준입력이 경사함수이면 정상상태오차가 0이 될 것이다. 그러나 시스템이 이제 3차 시스템이므로 원래의 2차 시스템보다 덜 안정하게 되든지 파라미터 K_p와 K_i를 적절히 선택하지 못하면 불안정하게 된다.[32]

PD제어를 갖는 1형태인 시스템의 경우에는 경사오차상수 K_v가 K_p에 비례하므로 K_p의 값이 중요하며 따라서 입력이 경사함수일 때 정상상태오차는 K_p에 반비례한다. 반면에 K_p가 너무 크면 시스템이 불안정하게

된다. 마찬가지로 0형태인 시스템에 대해서도 계단입력에 따른 정상상태 오차가 K_p에 반비례한다.

1형태인 시스템을 PI제어기를 이용하여 2형태로 전환할 때 K_p는 더 이상 정상상태오차에 영향을 주지 않으며 경사입력을 갖는 일정한 시스템에서는 오차가 항상 영이다. 다음 과제는 과도응답이 충족되도록 K_p와 K_i를 적당히 조합하여 선택하는 것이다.

*3.2 PI*제어의 시간영역 해석과 설계

식(3-1)의 PI제어기에 대한 극-영점 배치는 그림 3.3에 나타내었다. PI제어기는 안정도의 대가로 정상상태오차를 개선하는 것처럼 보인다. 그러나 $G_c(s)$의 영점의 위치가 적당하게 선정된다면 감쇠와 정상상태오차 모두를 개선할 수 있다. PI제어기는 본질적으로 저역통과필터이므로 보상된 시스템은 일반적으로 느린 상승시간과 긴 정정시간을 갖는다. PI제어를 설계하는 실현 가능한 방법은 영점이 상대적으로 원점에 가깝고 공정의 가장 의미 있는 극에서는 멀리 떨어지도록 $s=-K_i/K_p$에 선택하며 K_p와 K_i의 값들은 모두 비교적 작게 선택하는 것이다.

*3.3 PI*제어의 주파수영역 해석과 설계

주파수영역 설계를 위한 PI제어기의 전달함수는 다음과 같다.

$$G_c(s)=K_p+\frac{K_i}{s}=\frac{K_i(1+\frac{K_p}{K_i}s)}{s} \qquad (3\text{-}7)$$

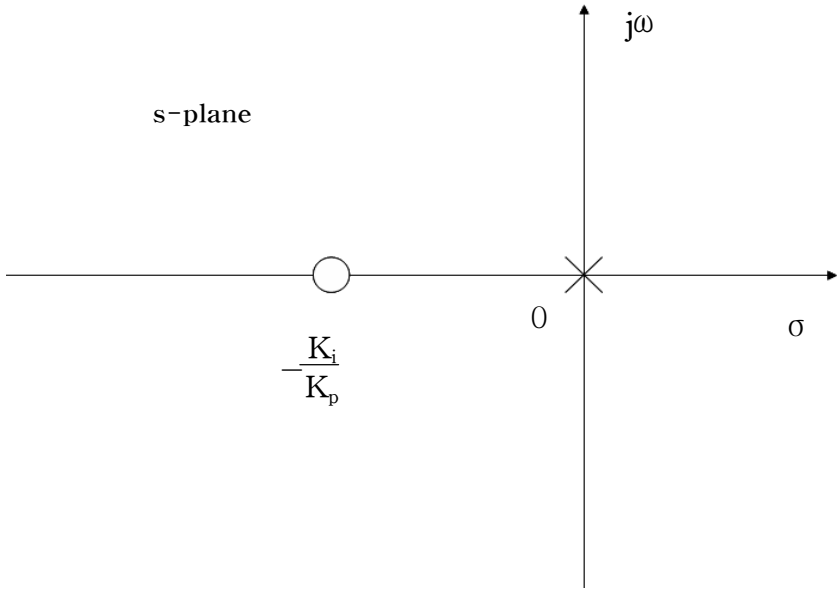


Fig 3.3 Pole-zero placement of PI controller

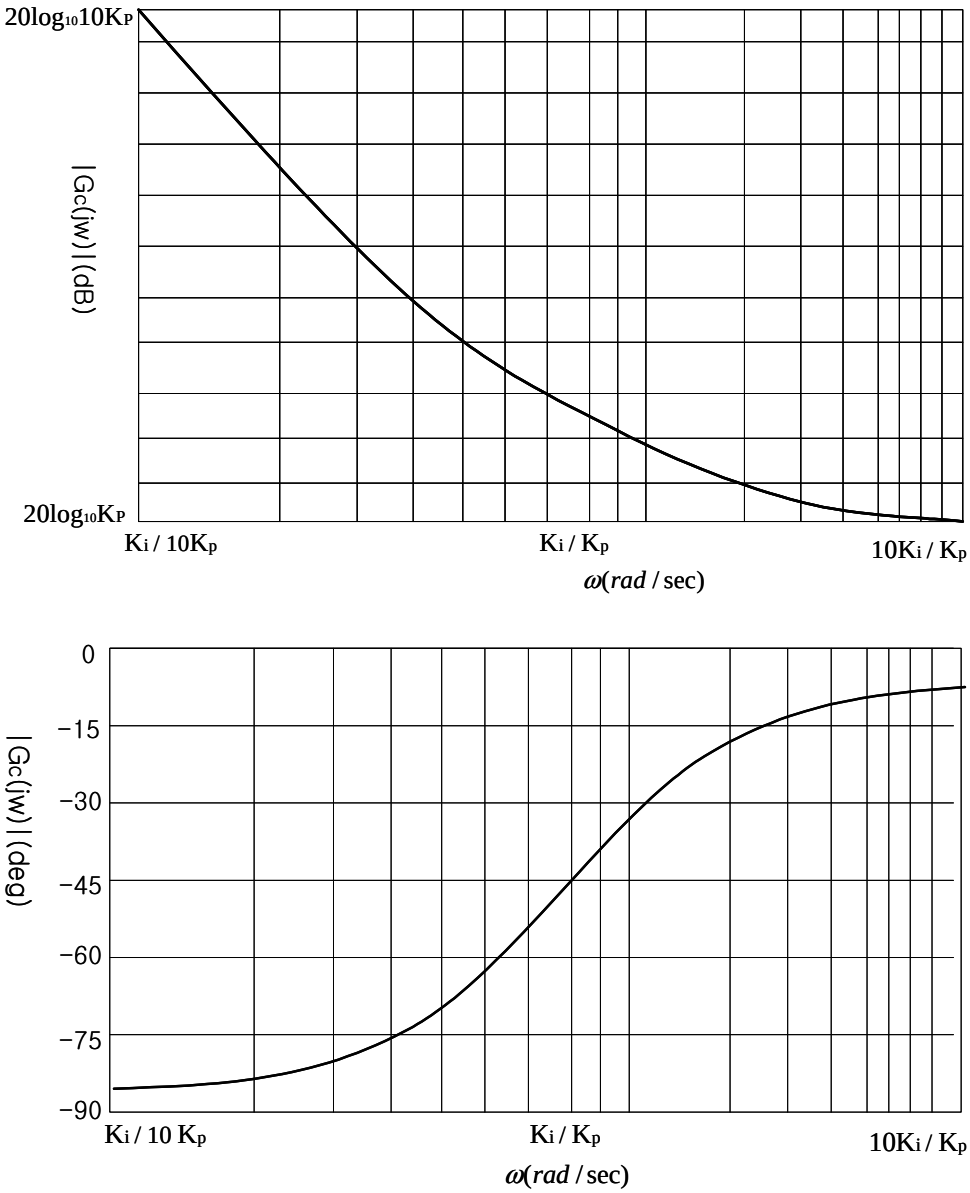


Fig 3.4 Bode diagram of PI controller $G_c(s)=K_p+\frac{K_i}{s}$

$G_c(j\omega)$의 Bode 선도는 그림 3-4와 같다. $\omega=\infty$에서 $G_c(j\omega)$의 크기는 $20\log_{10}K_p$[dB] 이며 이는 K_p가 1보다 작으면 감쇠를 나타낸다. 이런 감쇠는 시스템의 안정도를 개선하는 데 도움이 될 수 도 있다. $G_c(j\omega)$의 위상은 항상 음의 값을 가지며 이는 안정도에 바람직하지 못하다. 따라서 제어기의 절점 주파수 $\omega=K_i/K_p$를 대역폭의 요구조건이 허용하는 대로 왼쪽으로 멀리 설정하여 $G_c(j\omega)$의 지상특성이 시스템의 위상이득을 손상하지 않도록 선택해야 한다.

주어진 위상여유를 실현하기 위한 PI제어의 주파수영역 설계과정은 다음과 같다.

1. 보상되지 않은 시스템의 전방경로 전달함수 $G_p(s)$의 Bode 선도를 정상상태의 성능 요구조건을 만족시키는 루프이득을 갖도록 만든다.
2. Bode 선도로부터 보상되지 않은 시스템의 위상여유와 이득여유를 구한다. 주어진 위상여유 요구조건에 따라 이 위상여유에 대응하는 새로운 이득교차 주파수 ω'_g을 Bode 선도 상에서 찾는다. 보상된 전달함수의 크기선도는 필요한 위상여유를 실현할 수 있도록 이 새로운 이득교차 주파수에서 0[dB]축을 지나도록 한다.
3. 보상되지 않은 전달함수의 크기곡선을 새 이득교차 주파수 ω'_g에서 0[dB]로 감소시키기 위해 PI제어기는 새 이득교차 주파수에서 크기곡선의 이득과 동일한 크기의 감쇠를 마련해야 하며 다음과 같이 놓는다.

$$|Gp(j\omega'_g)|dB=-20\log_{10}K_p[dB] \qquad K_p<1 \qquad (3\text{-}8)$$

$$K_p = 10^{-|G_p(j\omega'_g)|_{dB}/20} \qquad K_p < 1 \qquad (3\text{-}9)$$

K_p의 값이 결정되면 다음에는 설계를 완성하기 위해 적당한 K_i의 값을 선택하는 것이 필요하다. 지금까지는 이득교차 주파수를 $G_c(j\omega)$의 크기를 감소시켜 ω'_g에 위치하도록 할지라도 PI제어기가 원래의 위상에 영향을 주지 않는다고 가정하였다. 그러나 그림 3.4에서 볼 수 있듯이 PI제어기의 감쇠성질이 위상여유를 저해하는 지상을 초래하므로 위의 가정은 불가능하다. 절점 주파수 $\omega = K_i/K_p$를 ω'_g아래로 멀리 위치하도록 한다면 PI제어기의 지상은 ω'_g 근처에서 보상된 시스템의 위상에는 거의 영향을 미치지 않는 것이 분명하다. 반면에 K_i/K_p의 값을 너무 작게 하면 시스템의 대역폭이 너무 낮게 되어 상승시간과 정정시간이 너무 길어지는 원인이 된다. 일반적으로 K_i/K_p를 ω'_g 아래로 최소한 10배 또는 때에 따라 20배 정도의 주파수에 대응하도록 한다. 즉 다음과 같이 정한다.

$$\frac{K_i}{K_p} = \frac{\omega'_g}{10} \quad \text{rad/sec} \qquad (3\text{-}10)$$

이와 같은 지침 내에서 K_i/K_p는 설계자의 재량 아래 BW에 미치는 영향과 op 증폭회로의 실제적 구현을 고려하여 결정되어야 한다.

4. 성능사양을 모두 만족시키는가를 보상된 시스템의 Bode 선도를 이용하여 검토한다.

5. PI제어기에 필요한 전달함수를 얻기 위해 K_i와 K_p의 값을 식(3-7)에 대입한다.

제어공정 $G_p(s)$가 0형태라면 K_i의 값을 경사오차상수에 대한 요구조건에 따라 선택해도 좋으며, 이 경우에는 파라미터 K_p만 결정하면 된다. 일정한 범위의 K_p 값에 대하여 폐루프 시스템의 위상여유, 이득여유, M_r 및 BW를 계산하여 최적의 K_p값을 선정한다.

이상의 논의를 근거로 적절히 설계한 PI제어기의 장단점은 다음과 같다.

1. 감쇠를 개선하고 최대오버슈트를 줄인다.
2. 상승시간이 증가한다.
3. BW를 감소시킨다.
4. 이득여유와 위상여유 및 M_r을 개선한다.
5. 고주파수 잡음을 여과한다.
6. 제어기의 회로 구축시에 커패시터가 과도학 크지 않도로 하는 K_i와 K_p의 적당한 조합을 선택하는 문제는 PD 제어기의 경우보다 더욱 어렵다.

Ⅳ. 유전자알고리즘

4.1 유전자 알고리즘의 진화

유전자 알고리즘은 1970년대에 미국의 Holland와 그의 문하생들과 동료들에 의하여 체계화된 것으로 확률적 탐색이나 학습 및 최적화를 위한 한 가지 기법이라고 할 수 있다. 처음 이들 유전자 알고리즘, 유전자 프로그래밍, 진화전략분야는 독립적으로 제창되었으나 서로 간의 분명한 선을 그을 수 있는 형태가 아니고 현재에 이르러 서로 밀접한 관계에 이르게 되었다.[33]

유전자 알고리즘상의 연구의 중심 논제는 견고성(robustness)인데, 많은 다른 환경에서 생존하기 위해 필요한 효율과 실효간의 균형이다. 인공 시스템에서의 견고성의 함축적 의미는 다양하다. 인공 시스템은 좀 더 견고하게 만들어질 수 있다면, 재설계는 감소되거나 제거되어 진다. 적응의 수준이 달성되어 진다면, 기존의 시스템은 그들의 기능을 좀 더 잘 수행할 수 있다. 인공적인 시스템의 설계자는 생물학적 시스템의 견고성, 효율성, 유연성에 경이로울 뿐이다. 자기보수, 자기조정, 재생산에 대한 특징은 생물학적 시스템에서의 규칙이다. 반면 아주 정교한 인공 시스템들에서는 거의 존재하지 않는다. 유전자 알고리즘은 복잡한 공간에서의 견고한 탐색을 제공하는 것이 이론적으로나 경험적으로 입증되었다.[34]

유전자 알고리즘은 두 보모의 유전자로부터 그들 자손의 형성하는 유전생식과 자연환경에서 일어나는 진화원리를 모방하고 있다. 자연계에 있는 생물의 진화과정에 있어서, 어떤 세대를 형성하는 개체들의 집합, 즉 개체군 중에서 환경에 대해 적합도가 높은 개체가 높은 확률로 살아남아 재생 할 수 있게 되며, 이 때 교배(crossover) 및 돌연변이(mutation)로써 다음 세대의 개체군을 형성하게 된다. 유전자 알고리즘에서 개체의 수를 개체군의 크기라고 한다. 각각의 개체는 염색체를 가지고 있고, 염색체는 복수개의 유전자의 집합으로 구성된다. 유전자의 위치를 유전자좌(locus)

라고 하고 유전자가 취하게 되는 유전자의 후보를 대립유전자(allele)라 한다. 이와 같이 유전자에 의해 결정되는 개체의 형질을 표현형(phenotype)이라 하고 이에 대응되는 염색체의 구조를 유전형(genotype)이라 한다. 표현형이 여러 개의 유전자좌의 영향을 받아 복잡한 형태가 결정되는데 이것을 에피스타시스(epistasis)라고 한다. 또한 표현형을 유전형으로 바꾸는 것을 부호화 그 역을 복호화라고 한다. 이와 같이 유전자 알고리즘은 생물의 진화과정을 인공적으로 모델링한 알고리즘이라 할 수 있다.[35]

4.1.1 유전자 알고리즘의 동작 및 특성

기본적인 유전자 알고리즘의 순서도는 다음과 같다.

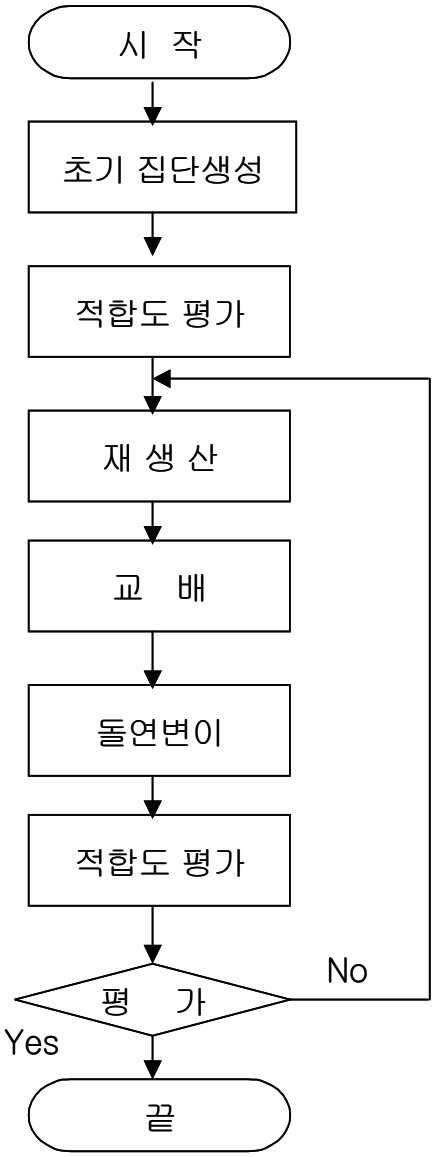


Fig 4.1 Flow chart of genetic algorithm

유전자 알고리즘은 전통적인 탐색 및 최적화 방법들과 여러 가지 면에서 차이를 가지고 있다. 구체적인 차이점들은 다음과 같다.[36~38]

① 파라미터를 직접 이용하기보다는 적절히 코딩하여 이용한다.

유전자 알고리즘은 문제 공간을 탐색하기 위해 파라미터를 이용하기보다는 코드화된 파라미터를 이용하여 염색체(chromosome)라 하며 유전 연산자가 코드화된 파라미터에 적응하여 원하는 해 개체군(population of solution space)을 얻게 되고, 개체군을 디코딩하여 원하는 해 파라미터를 구한다.

② 하나의 해를 다루기보다 집단을 다룬다.

유전자 알고리즘은 하나의 파라미터가 아닌 코드화 된 파라미터의 개체군을 가지고 문제공간을 탐색한다. 즉 보다 넓은 공간을 탐색하게 되며 국부점(local minima)에 빠지지 않도록 된다.

③ 탐색에 비용 정보를 이용하며, blind search를 한다.

유전자 알고리즘은 문제영역에 특별한 정보가 필요치 않다. 간단한 예로 calculus-based 탐색방법은 문제의 미분량이나 연속성 등의 수학적 정보를 알아야 하나 유전자 알고리즘은 단지 문제 공간에 대한 인공 염색체의 적합도만이 필요할 뿐이다.

④ 연산자로서 유전 연산자들을 사용한다.

유전자 알고리즘은 진화를 위해 교배나 돌연변이 등과 같은 자연현상을 모방한 유전 연산자들을 사용한다.

⑤ 결정론적인 규칙이 없고 확률적 연산자를 사용하여 수행된다.

유전자 알고리즘은 확률적 연산자를 사용한다. 단순 유전 알고리즘에서는 교배와 돌연변이 연산자이며 부모의 유전적 형질을 자손에게 확률적으로 전하는 작업을 한다.

이와 같은 특징으로 인해 다른 탐색 또는 최적화 방법 중 하나인 계산에 의존한 방법(calculus-based method : hill-climbing)에 비하여 전역적 해(Global solution)를 구할 가능성이 높으며 다른 여러 탐색 방법에 비하여 효율적이다.

4.2. 유전자 알고리즘의 연산자

4.2.1 복제와 교배

복제는 각각의 문자열이 가지는 적합도에 따라 그 문자열을 재생산하는 과정이며, 이 과정은 적합도가 높은 개체일수록 다음 세대에 더 많은 자손을 가질 확률이 높음을 의미한다. 이것은 주어진 환경에 더 잘 적응하는 개체만이 살아남는다는 자연선택의 원리를 담고 있다. [39~40]

교배는 2개의 개체 간에 염색체를 부분적으로 서로 바꿈으로써 새로운 개체를 생성하는 것이다. 이 때 부모의 형질이 자손에게 적절히 계승되어야 하며 이것을 형질 유전성 또는 교배 근접이라 한다.[41]

가. 단순교배(simple crossover)

다음과 같이 부모 염색체 p_1, p_2가 있고 교배점(crossover site)을 3으로 했을 때, 생성되는 자식은 s_1, s_2이 같이 된다.

$$p_1 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \qquad s_1 = a_1 a_2 a_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$$

$$\updownarrow \qquad\qquad \Rightarrow$$

$$p_2 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \qquad s_2 = a_1 a_2 a_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$$

나. 복수점 교배(multi-point crossover)

단순교배와 같으나 교배점이 2개 이상이다. 다음은 교배점이 2, 5, 7의 세 개로 했을 때 생성되는 결과를 나타낸다.

$p_1 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$ $\qquad$ $s_1 = a_1 a_2 a_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$

$\updownarrow \quad \updownarrow \quad \updownarrow$ $\qquad$ $\Rightarrow$

$p_2 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$ $\qquad$ $s_2 = a_1 a_2 a_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$

다. 균일교배(uniform crossover)

이것은 각 유전자가 독립적으로 교환될 수 있도록 하는 것으로서 다음과 같이 마스크를 씌워 비트가 0일 경우에만 유전자를 교환한다.

$p_1 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$ $\qquad$ $s_1 = a_1 a_2 a_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$

m = 1 0 0 1 1 1 0 1 0 0 $\qquad \Rightarrow \qquad$ m = 1 0 0 1 1 1 0 1 0 0

$p_2 = b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$ $\qquad$ $s_s = b_1 \underline{a_2 a_3} b_4 b_5 \underline{b_6 a_7} b_8 \underline{a_9 a_{10}}$

라. 부분일치교배(partically matched crossover : PMX)

보통 TSP 문제에서 아래와 같이 A와 B의 두 개의 염색체를 교배할 경우 2개의 교배점을 잡고 그 사이의 중간부분을 일치 시켜 교환한 후 나머지 부분의 그 결과에 따라 중복되는 부분을 피하여 조절한다.

A = 984|567|13210 $\rightarrow$ A′ = 984|2310|1657

B = 871|2310|9546 $\rightarrow$ B′ = 8101|567|9243

마. 순서교배(ordered crossover : OX)

TSP 문제에서 순서 교배의 예는 다음과 같다.

A = 984|567|13210

B = 871|2310|9546

$\rightarrow$ B = 8H1|2310|9H4H

$\rightarrow$ B′ = 2310|HHH|9481

A′=567|2310|1984

B′=2310|567|9481

2개의 부모개체를 선택하여 개체 문자열의 임의의 위치에서 두 개의 개체를 서로 바꾸는 과정이다. 다음 그림은 교배의 한 예를 설명한 것이다.

	1	0	1	1	0	1	1	0
before crossover	1	1	0	0	1	0	0	0
after crossover					↑			
	1	1	0	0	0	1	1	0
	1	0	1	1	1	0	0	0

Fig. 4.2 Crossover

위 그림에서 보듯, 2개의 부모 개체에서 임의의 bit위치를 선정하여, 그 이후의 문자열을 서로 바꾸는 과정이다. 교배 이전의 첫 번째 문자열은 1 0 1 1 0 1 1 0이고 두 번째 문자열은 1 1 0 0 1 0 0 0이다. 4번째 bit 위치에서 교배가 일어난다고 임의로 설정하면, 교배 이후의 두 개의 문자열은 각각 1 1 0 0 0 1 1 0 과 1 0 1 1 1 0 0 0 으로 되는 것이다.

이 과정은 교배확률을 가지는 확률적인 과정이지만 개체간의 정보를 교환하고 문자열의 좋은 부분을 결합하는 효과적인 방법이다.

유전자 알고리즘에서 교배의 가장 큰 독특한 주요 특징은 교배 연산자의 사용이라고 말할 수 있다. 일점교배는 가장 단순한 형태이다. 단 하나의 교배 위치가 임의로 선택되고 두 부모들의 교배 위치 이후의 부분들이 두 개의 자손들을 형성하기 위해 교환된다. 물론, 여기서 이 아이디어는 다른 문자열 상의 구성부들(스키마)을 재결합시키는 것이다. 하지만, 일점교배는 몇 가지 단점들이 있다. 한 가지는, 이것이 모든 가능한 스키마를 결합시키지는 않는다는 것이다. 예를 들면, 11**11*1을 형성시키기 위해 1*****1과 ****11**를 일반적으로 훼손될 가능성이 있다. 교배에 의해 생성되거나 파괴될 수 있는 스키마들은 염색체 내의 비트들의 위치에 강하게 의존한다. 일점교배는 짧고, 낮은 순위의 스키마들이 문자열의 기능적인 구성부 라는 가정을 하지만, 비트들의 어떤 순위가 기능적으로 관련된 비트들을 같이 그룹화 할 것인지를 일반적으로 미리 알지 못한다.

대체로 이들 중 상당 부분이 다른 형태의 부호화에도 또한 적합하다. 몇몇 부호화 형태들은 특별하게 정의된 교배 연산자나 돌연변이 연산자들을 요구한다.

이와 같이 교배 연산자를 사용하는 주된 이유가 고도로 적합한 스키마들을 결합시킬 수 있는 능력이 있기 때문이다. 유전자알고리즘을 이해하기 위한 분석도구로서 스키마들의 타당성에 대해 우리가 확인해 왔던 몇 가지 도전할 만한 것들이 주어졌을 때, 교배가 전적으로 다른 어떤 이유 때문에 실질적으로 유용하다는 가능성은 알 수 있다.

4.2.2 돌연변이와 역위

개체의 각 유전자좌의 유전자에 대하여 일정한 돌연변이 확률 (p_m)을 적용하여 대립 유전자의 값으로 바꾸는 것이다. 개체에 근접한 새로운 개체를 생성하는 국소적인 랜덤 탐색의 일종이다. 또한 집단에서 잃어버린 유전형질을 복구하여 다양성을 유지하기 위한 수단으로도 사용된다. 이때 전형적인 돌연변이 확률은 0.05이하이다.

다음은 돌연변이의 적용례이다. 세 번째 비트인 a_3가 A_3로 바뀌었다.

$$p = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \quad \rightarrow \quad s = a_1 a_2 \underline{A_3} a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$$

하나의 개체에서 임의로 어떤 bit를 선택하여 그 bit가 1이면 0으로 0이면 1로 바꾸는 과정이다. 돌연변이는 위에서 설명한 복제와 교배의 과정에서 생길 수 있는 중요한 유전 정보의 손실을 막고 탐색의 방향이 지역적 극값으로 향하는 것을 막아주는 역할을 한다.

유전자알고리즘에서는 교배가, 어느 특정한 위치에 개체집단이 연속적으로 고정되지 않도록 보장하는 등의 배후 역할 부분을 담당하는 돌연변이와 더불어, 유전자알고리즘 내의 변화와 혁신의 도구의 역할을 한다. 이것은, 랜덤 돌연변이만이 변화의 유일한 근원인 진화 프로그래밍과 진화전력의 초기 형태 같은, 다른 진화연산 방법과는 다르다.

그러나 무엇보다도 중요한 점은 교배 또는 돌연변이를 선택하는 것보다는 차라리 교배, 돌연변이, 선택 사이의 균형을 이루는 것이다. 옳은 균형은 또한 적합도 함수와 부호화의 세부사항들에 의존한다. 더욱이, 교배와 돌연변이는 실행 과정 중에 상대적인 유용성이 변화한다.

다음과 같이 두 개의 역위 점을 구하여 그 사이의 비트들의 순서를 반대로 바뀐다.

$$p = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \quad \rightarrow \quad s = a_1 a_2 \underline{a_8 a_7 a_6 a_5 a_4} a_9 a_{10}$$

↑

그 밖에 치환, 중복, 추가, 제거등의 연산자가 사용되고 있다.

4.3 유전자 알고리즘의 적합도 평가(fitness evaluation)

한 생명의 생존능력은 그 환경에 대한 적합도(fitness)를 반영한다. 환경에 대한 생물의 적합도는 장점을 의미하고 크면 클수록 더 많은 보상을 받듯이, 유전자 알고리즘에서도 적합도가 큰 개체가 더 많은 보상을 받을 수 있도록 배려되어야 한다.

목적 함수 즉, 최적화 하고자 하는 함수는 각 개체의 적합도를 평가하는 기반이다. 그러나 목적함수 값의 범위는 문제마다 다르기 때문에 보통 정해진 구간 사이의 양수 값을 갖도록 표준화된 값을
사용한다. 즉, 표준화하기 이전의 적합도의 값을 raw fitness라고 하며 표준화되어서 실제로 개체 선택의 기준이 되는 함수를 적합도 함수라고 한다. raw fitness를 표준화하는 방법은 대표적으로 3가지가 있다.

4.3.1 선형 표준화(linear scaling)

선형 표준화하는 순서는 아래와 같다.

<1단계> 개체군의 평균 적합도를 $\bar{f}$, 최대 적합도를 f_{max}, 최소 적합도를 f_{min}이라 하고 c=2.0로 한다.
(c는 최고 좋은 개체가 다음 세대에 재생할 개체수의 기대치를 말하는 것으로, 보통 개체군의 크기가 50~100일 경우 c=0.2~2 정도로 한다.)

<2단계> $f_{min} > \frac{c \cdot \bar{f} - f_{max}}{c - 1.0}$ 이면 <3단계>로 가고 그렇지 않으면 <4단계>로 간다.

<3단계> $a=\frac{(c-1.0)\cdot\bar{f}}{f_{max}-\bar{f}}$, $b=\frac{\bar{f}(f_{max}-c\cdot\bar{f})}{f_{max}-f}$ 으로 하고 <5단계>로 간다.

<4단계> $a=\frac{f}{f-f_{max}}$, $b=\frac{f_{min}\cdot\bar{f}}{f-f_{max}}$ 로 하고 <5단계>로 간다.

<5단계> $i=1,2,\cdots,N$에 대하여 $f_i=af_i+b$ 로서 선형 스케일링을 행한다.

4.3.2 σ절단

계산된 적합도의 표준편차를 고려하는 방법으로서 개체군의 평균 적합도의 값을 $\bar{f}$, 표준편차를 σ라 할 때 다음의 식에 의해 스케일링을 행한다.

$$f_i=f_i-(\bar{f}-c\cdot\sigma) \qquad (4\text{-}1)$$

상수 c는 표준편차에 곱해주는 수로서 보통 1에서 3사이의 값을 취한다. 여기서 $f_i<0$ 이면 $f_i=0$으로 한다.

4.3.3 거듭제곱 표준화

Gillies는 machine vision 응용문제에 대하여 k=1.005로 하여 다음과 같이 거듭제곱의 형태로 스케일링을 하였다.

$$f=(f_i)^k \qquad (4\text{-}2)$$

일반적으로 k의 값은 문제에 의존적이며 적당한 범위 내에서 크기를 변화시켜가며 좋은 결과가 나오는 값을 사용하게 된다.

4.3.4 정지 조건

집단이 모의진화를 거듭하면서 해 쪽으로 수렴해갈 때 언제 정지할 것인지를 결정하는 것은 대단히 중요하다. 두 가지 방법이 자주 이용되는데, 하나는 현재의 세대 또는 반복횟수가 미리 지정한 값(최대 세대수)에 도달하면 정지하는 것이다. 개체군이 해 가까이로 수렴해가면 개체들의 대부분은 한 세대에서 다음 세대로 아무런 변경 없이 생존해간다. 이 경우에는 모든 개체들의 적합도를 평가할 필요가 없게 되고, 지정한 세대에 도달하면 종료한다. 그러나 이 방법은 사용자가 탐색공간의 특성을 사전에 잘 알고 있다는 가정 하에서 성립한다. 탐색공간의 특성은 탐색의 길이에 영향을 주기 때문에 최대 세대수를 임의적으로 정하기보다는 주어진 문제에서 시행착오적으로 결정하는 것이 바람직하다. 또 다른 하나는 다시 개체의 유전자형에 기초한 것과 표현형에 기초한 것으로 나눌 수 있다. 전자는 매 세대마다 염색체들의 유전자 수를 조사하여 수렴성을 판단하는 것으로서, 수렴된 유전자의 수가 총 유전자의 수의 일정 비율을 넘게 되면 탐색을 종료한다. 후자는 매 세대마다 얻어지는 잠정적인 해로부터 수렴성을 조사하는 것이다.

4.3.5 알고리즘 제어 파라미터

유전 알고리즘에 의한 탐색 또는 최적화 문제의 해결에 있어서 미지의 영역에 대한 탐색과 획득한 정보의 유효한 이용의 조화가 중요하다. 획득한 정보의 유효한 이용은 기존의 등고선법과 유사해 지며, 미지의 영역에 대한 탐색이 강조 될수록 랜덤 탐색과 같은 특성을 나타내게 된다. 유전자 알고리즘은 이러한 두 가지의 조건을 함께 제어할 수 있는 매력적인 알고리즘이다. 이를 제어하기 위한 파라미터에는 여러 가지가 있으나 중

요한 것으로 3가지를 둘 수 있다. 개체군의 크기(population size : M), 교배 확률(probability of crossover : p_c), 돌연변이 확률(probability of mutation : p_m) 이다.

일반적으로 큰 p_c와 p_m의 값은 알고리즘의 exploration 능력을 향상시킴으로 진화의 초기에 적합도가 높은 탐색공간을 찾는데 유리하게 작용하지만, 동시에 explortation 능력을 저하시킴으로써 어느 정도 좋은 해를 찾은 후에는 이 탐색공간 내에서 최적해를 수렴하는 데 있어서는 수렴속도를 저하시키는 요인으로 작용할 수 있다. 작은 p_c와 p_m의 값은 반대의 특성을 나타낸다. 한편, 개체군의 크기 M이 작으면 적합도 계산에 필요한 시간을 절약할 수 있으나 개체간의 다양성의 빠른 손실로 인해 최적의 해를 구하기 전에 수렴할 위험성을 내포한다. 반면에 개체군의 크기가 크면 최적해에 도달할 확률은 높으나 많은 기억용량과 계산시간을 필요로 한다. 이 둘의 성능 평가 항목을 모두 만족시켜 주는 최적의 개체군의 크기를 정하는 방법은 문제의 성격과 다른 제어 파라미터들의 값에 따라 다르다.

4.4 유전자 알고리즘의 특성

유전자 알고리즘은 지금까지의 방법으로서는 해결이 곤란한 문제에 대해서 실용상의 최적해를 랜덤하고 빠르게 찾는 방법으로서 사용되어 왔다. 유전자 알고리즘은 종래의 검색 알고리즘과 달리 검색 공간중 하나의 점뿐만 아니라 많은 점을 동시에 고려하기 때문에 국부 최소점 점에 수렴되는 현상을 극복할 수 있다.[42~43]

유전자 알고리즘은 모든 해들이 유전자 구조로 만들고, 입력값 집단을 랜덤하게 선택한 후 이를 보다 최적 해에 가까운 집단으로 진화시키는 세대교체과정을 반복수행 한다.

다음과 같이 4가지 방식에 있어 좀 더 일반적인 최적화와 탐색 절차들과 구분된다.

(가) 유전자 알고리즘은 파라미터 그 자체가 아닌, 파라미터 집합을 코딩하여 작동한다.

(나) 유전자 알고리즘은 단일점이 아닌 점들의 집단으로 부터 탐색한다.

(다) 유전자 알고리즘은 도함수나 다른 보조적인 지식이 아닌, payoff 정보를 사용한다.

(라) 유전자 알고리즘은 결정적인 규칙들이 아닌, 확률적인 천이규칙들을 사용한다.

Table 4.1 Compare the language of the nature with genetic algorithm

자 연	유전자 알고리즘
염색체(chromosome)	string
유전자(gene)	feature, character, detector feature
대립형질(allele)	value
인자위치(locus)	string position
인자형	structure
표현형(phonotype)	parameter set, alternative solution, a decoded structure
epistasic	onlinearity

4.5 유전자 알고리즘 적용

유전자 알고리즘 적용은 초기의 모집단을 생성, 평가, 진화, 최우수집단 갱신 등의 반복과정을 통해 수행된다. 로봇의 위치를 초기 노드로 하여

랜덤하게 초기의 모집단을 설정한다. 생성된 모집단에 대한 평가를 하여 생성된 4개의 경로집합을 적응도가 높은 순서로 정렬한다. 다음 세대를 위하여 두 가지의 과정을 수행하다. 첫 번째 과정은 교차이고, 두 번째 과정은 돌연변이이다. 먼저 교차과정에 대하여 알아보면, 기존의 최우수 집단과 4개의 교차된 집단 중 첫 번째 집단을 교차시켜 다음 세대 집단 0, 1을 만들고, 두 번째 집단과 교차시켜 다음세대 집단 2, 3을 생성한다. 다음으로 돌연변이의 과정을 보면 1/3 확률값 으로 랜덤하게 발생하는 돌연변이 플레그에 의해 동작되며 방법으로는 다음 기존의 최우수집단과 4개의 우수집단 중 첫 번째 집단에서 임의의 점을 택하고, 그 점부터 랜덤하게 경로를 설정하여 새로운 집단을 만든다. 또한 생성된 최우수집단은 기존의 최우수집단과 비교하여 적응도를 비교하여 최우수집단을 갱신한다. 생성된 집단은 재평가를 통하여 우수집단을 선택하여 교차하는 과정을 반복한다. 일정 횟수를 반복하여 가장 좋은 경로를 최단경로로 선택한다.

4.5.1 초기 모집단 생성

유전자 알고리즘을 적용하기 위하여 먼저 모서리 경로탐색에 의해 얻어진 가능경로의 그래프를 얻는다. 입력된 가능경로를 바탕으로 로봇의 위치를 초기 노드로 하여 랜덤하게 모집단을 설정한다. 식(3-3)을 사용하여 초기 모집단을 얻는다. 초기의 모집단 개수(k)는 4로 설정하였다.

$$\mathrm{Go} = \mathrm{G.G_i^k} = \begin{cases} 0 & \mathrm{i} = 0 \\ \mathrm{NC}_{\mathrm{l}(\mathrm{N_m})} & 1 < \mathrm{i} < \mathrm{NGO}^{\mathrm{k}} \end{cases},$$

$$\mathrm{l} = \mathrm{random}\,(\mathrm{Npath}_{\mathrm{G_{i-1}^k}}),\ \mathrm{m} = \mathrm{G_{i-1}^k},\ \mathrm{k} = 0 \sim 3 \qquad (4\text{-}3)$$

$\mathrm{NC}_{\mathrm{l}(\mathrm{N_m})}$: 노드 $\mathrm{N_m}$의 NC 집합 중 l번째 원소

NGO^k : k번째 경로집합의 총 경로 노드 수

G_i^k : k번째 경로집합의 i번째 경로노드

$Npath_{G_{i-1}^k}$: k번째 경로집합의 i－1 번째 경로노드의 이동가능 노드 수

GO 집합은 NODE 집합에서 원소들을 랜덤하게 선택하는 것과 같다. 즉 노드 N_i 의 Npath 사이의 값 중에 랜덤한 수 1을 생성하여 N_i와 연결 가능한 노드에 대한 정보집합(NC)에서의 *l*번째 노드번호를 참조하여 G^k 집합의 i번째 원소로 설정한다. 다음 G^k 집합의 i＋1번째 원소는 G^k 집합의 i번째 원소에 대한 노드 정보를 참조하여 설정한다. 두 노드간 무한 루프에 빠지는 것을 방지하기 위하여 p 노드에서 q 노드로 이동하였을 경우에 p와 q의 연결을 끊는다. 즉 p노드에서 이동 가능 노드번호에서 q번을 삭제하고, q 노드에서 이동가능 노드번호 중에서 q번을 삭제하였다.

G^k의 원소인 노드번호가 목표점의 노드번호와 같거나 $Npath_{G_{i-1}^k}$의 값이 0일 경우에는 G^k 집합에 대한 경로설정을 종료한다.

4.5.2 평 가

얻어진 경로들에 대한 평가를 실시한다. 본 논문에서 사용한 평가 함수는 식 (4-4), (4-5)와 같다. 평가함수 fit는 거리함수에 반비례하며, 최종 노드 값이 목표점의 노드 값인지를 판별하여 아닐 경우는 거리함수에 가중치 NGBEST를 추가하였다. NGBEST는 목표점에 도달하지 않은 경로 중 가장 많은 노드를 이동한 경로의 이동노드 수이다. 적합도 평가는 경로들의 거리합계로서 계산되며 적합도가 높은 것이 좋은 경로이다. 또한 목표점까지 경로가 설정되지 못한 경우는 가장 많은 경로를 거친 경로

가 좋은 경로이다.

$$Dist^k = \begin{cases} \sum_{i=1}^{NGO^k} & \sqrt{(x_{Ni} - x_{Ni-1})^2 + (y_{Ni} - y_{Ni-1})^2} \\ & ,if\ (x_{NGO^k-1} = x_{goal} \text{ and } (y_{NGO^k-1} = y_{goal}) \\ \sum_{i=1}^{NGO} & \sqrt{(x_{Ni} - x_{Ni-1})^2 + (y_{Ni} - y_{Ni-1})^2} \\ & + (NGBEST - NGO^K) * K_p \\ & ,\text{if } (x_{N_{NGO^k-1}}\ x \neq goal \text{ and } \neq y_{Gola}) \end{cases} \quad (4\text{-}4)$$

x_p : 비례상수(=25)

$$\text{fit}^k = \frac{100}{\text{Dist}^k} \quad (4\text{-}5)$$

4.5.3 저장 및 진화

평가함수의 값을 키 값으로 하여 내림차순으로 정렬한다. 가장 fit 값이 높은 경로가 SG^0 에 위치하게 되고, 그 다음으로 SG^1, SG^2, SG^3으로 위치하게 된다.

$$SGO = \text{sort}(GO)$$

SG^0와 BEST_GO의 적응도를 비교하여 좋은 것을 BEST_GO로 저장, 갱신한다.

$$\text{if } fit_{SG^P} > fit_{BEST-GO} \text{ then}$$

$$\text{begin}$$

$$exBEST_GO = BEST_GO$$

$$BEST_GO = copy(SG^0)$$

$$\text{end}$$

4.5.4 선택, 교차, 돌연변이

새로운 세대에 대한 경로를 위하여 SG^0 경로와 SG^1의 경로를 선택하고, BEST_GO와 교차 또는 돌연변이를 실시한다.

$$\begin{cases} cross(SG^0, exBest_GO & : CrossFlag = 0 \\ cross(SG^0 \cdot SG^1) & : CrossFlag = 1 \\ RanGO(SG^0 \cdot SG^1) & : CrossFlag = 2 \\ RanGO(SG^2 \cdot SG^3) & : CrossFlag = 3 \end{cases} \qquad (4\text{-}6)$$

교차는 CrossFlag에 의해 작업이 분리되었다. CrossFlag0, 1은 교차를 수행하여 새로운 경로집합을 얻을 경우에 설정되고, 2, 3은 교차 실패 또는 돌연변이를 수행할 경우 새로운 경로집합을 얻고자 할 때 설정된다. 식 (4-6) cross(A, B)함수는 경로 A와 경로 B에서 임의의 공통노드를 찾아 경로를 교차하여 새로운 경로를 생성하는 함수이며, RanGO(A, B)는 경로 A, B에서 임의의 노드로부터 이동경로계획을 다시 수행하여 새로운 경로를 얻는 함수이다.

식 (4-7)에 $cross(SG^1, SG^2)$함수에 대한 예를 나타내었다. 랜덤한

수 2를 얻었을 경우 SG^1는 1이므로 SG^2에서 1을 탐색한다. SG^1에서 1 이하의 노드 즉 {2,3,4,3,6}과 SG^2에서 4이하의 노드 {4,8,9,6}를 서로 비교하여 다음으로 같은 노드를 탐색한다. 노드 4가 같이 존재하고 있으므로 새로운 경로를 위하여 SG^1에서는 {1,2,3,4}와 SG^2 {1,4}가 서로 교차되어 새로운 경로 $NewG^1$과 $NewG^2$를 얻었다.

$$
\begin{aligned}
&SG^1 = 0,1,2,3,4,3,6 \\
&SG^2 = 0,1,4,8,9,6 \\
&\mathrm{cross}\,(SG^1, SG^2) \\
&NewG^1 = \{0,1,4,3,6\} \\
&NewG^2 = \{0,1,2,3,4,8,9,6\}
\end{aligned}
\tag{4-7}
$$

만약 SG^1과 SG^2의 교차를 위한 랜덤 Node를 찾지 못했을 경우와 돌연변이 프래그가 1일 경우에는 RandGo() 함수를 수행한다. RandGo(A, B) 함수는 랜덤 한 수 N1을 발생시키고 A와 B의 경로를 $N1$이후부터 초기 모집단 설정과 같은 방법으로 계속 진행시킨다. 식 (4-8)에 RandGo에 대한 함수의 예를 볼 수 있다. SG^1과 SG^2에 대한 같은 노드를 0번 노드 이외에는 찾을 수 없다. 그러므로 RandGo() 함수를 실행한다. 랜덤 한 수를 3을 선택하였을 경우 SG^1에서는 9번 노드에서 10번이 아닌 다른 로드 11번을 선택하여 경로 설정 과정을 수행한다. SG^2의 경우도 3번째 노드 2이하부터 랜덤 경로 설정을 진행한다.

$$
\begin{aligned}
&SG^1 = 0,8,9,10 \\
&SG^2 = 0,1,2,3,4,3,5,7,6 \\
&RandGo(SG^1, SG^2) \\
&NewG^1 = \{0,8,\underline{9,11,6}\} \\
&NewG^2 = \{0,1,\underline{2,5,9,6}\}
\end{aligned}
\tag{4-8}
$$

다음 세대로 얻어진 $NewG^k$ 의 경로를 다음 세대의 G^k로 이동시키고 max_ ≥neration 까지 평가, 선택, 교차, 돌연변이 과정을 반복수행하여 최적의 경로를 도출해 나간다.

REGENERATE : $G^k = NewG^k$

4.6 유전자 알고리즘 흐름도

가. 기본모델

적합도 비례선택, 룰렛 선택법 각 개체 S_i의 적합도 $f(s_i)(>0)$, i=1, …, N의 총합을 구해 각 개체 S_i 의 선택 확률을 다음 식(3-9) 과 같이 정의한다.

$$p_i = \frac{f(s_i)}{\sum_{j=1}^{N} f(s_i)} \qquad (4\text{-}9)$$

나. 순위 선택법(ranking selection)

적합도 비례선택과 기댓값 선택은 적합도 값에 의해 재생할 개체수를 결정하는데, 이는 합리적으로 생각되나 반드시 좋은 경우만 있는 것은 아니다. 예로써, 오차의 역수를 접합도로 선택을 할 경우 적합도가 특히 높은 개체가 발생하면 그 개체의 영향으로 전체의 개체가 유사해지고, 반대로 적합도의 차가 거의 없으면 개체의 선택 확률은 거의 균등해지는 문제가 발생해진다. 따라서 순위선택법은 적합도의 크기순서에 따라 순위를 매긴 후 순위에 따라 다음세대에 자손을 남길 확률을 결정하는 방법이다. Baker는 선형 순위선택법으로 각 개체의 순위가 i일때 개체 S_i 의 선택 확률을 다음 식(3-10) 같이 제안 하였다.

$$p_i = \frac{1}{N}(\eta^+ - (\eta^+ - \eta^-) \cdot \frac{i-1}{N-1}) \qquad (4\text{-}10)$$

여기서, $1 \le \eta^+ \le 2$, $\eta^- = 2 - \eta^+$로 하였고 η^+와 η^-는 각각 최대 기대

치와 최소 기대치를 나타낸다. 또한 Michalewiz는 비선형 순위선택법으로 다음과 같이 식(4-11)을 제안하였다.

$$p_i = c \cdot (1-c)^{i-1} \qquad (4\text{-}11)$$

여기서, c는 순위가 1인 개체가 선택될 확률을 나타낸다.

다. 토너먼트 선택법(tournament selection)

개체군 중에서 일정한 개수의 개체를 임의로 선택하여 그 다음 세대에 남기는 방법이다.

이것은 다음 세대의 개체 수가 만족할 때까지 반복한다.

라. 엘리트 보전 선택(elitist preserving selection)

확률에 따라 개체를 선택하여 교배 및 돌연변이의 결과로 특별히 좋은 해가 소실되는 것을 막기 위하여 가장 좋은 해를 보존하여 다음 세대에 남기는 방법이다. De Jong은 엘리트 선택을 다음과 같이 정의하였다.

$s^*(t)$ 를 세대 t의 가장 좋은 개체라 하자, 또 X(t+1)를 보통의 방법으로 생성할 때 X(t+1) 중에 $s^*(t)$ 가 존재하지 않으면 $s^*(t)$를 X(t+1)의 N+1 번째 개체로 추가한다.

V. 유전자알고리즘 *PI*제어기 설계

5.1 유전자 알고리즘(GA) PI 제어기

제어기를 설계하는 목적은 제어기를 이용하여 공정의 다양한 동특성을 보완하기 위해서이다.

본 논문에서는 교류서보전동기 속도제어를 위하여 PI제어기의 파라미터 설정을 조절하도록 GA를 이용한다.

정의된 성능지수 또는 평가함수를 설정하고 종래의 PI제어기에서 PI파라미터의 최적 설정치를 GA를 이용하여 구한다. 이를 GA-PI제어기라고 정의하고 비선형 최적화문제로써 고려되는 설정 점을 향하여 교류서보전동기의 속도제어를 수행한다.

GA-PI제어기의 기본적인 구조는 그림 5.1과 같다.

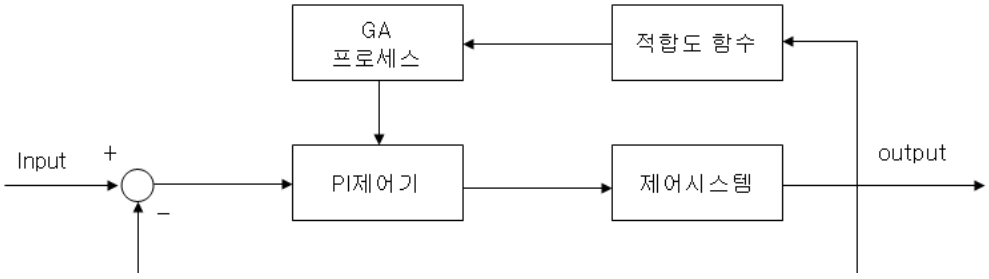


Fig. 5.1 Basic structure GA-PI controller

GA-PI의 단계별 실행과정은 다음과 같다.

(1) PI제어기의 속도와 전류제어기인 2개의 파라미터는 2개의 2진 비트 서브스트링으로 변환한다. 각 서브스트링의 길이는 l_i(i=1,2)이며 시간 t에서 일치하는 2진 서브스트링의 10진 값은 다음 식과 같다.

$$K(t) = K_{min} + \frac{binrep(t)}{2^{l_i}}[K_{max} - K_{min}] \qquad (5\text{-}1)$$

여기서 binrep(t)는 서브 스티링의 2진 값이며 파라미터 K는 폐구간 $[K_{max}, K_{min}]$의 영역에 있다고 가정한다. 이 서브스트링은 하나의 긴 스트링으로 코드화 된다. 스트링의 길이는 $L = \sum_{i=1}^{2} li$ 이며, 각 시그먼트는 2개의 파라미터중 하나의 값으로 표현된다.

(2) 코딩의 룰에 따라 랜덤하게 군집을 초기화한다.

(3) 현재의 세대에서 각 스트링은 실제 파라미터와 일치하게 디코드화 시킨다. 그런 다음, 이 파라미터는 조절 시스템으로 보낸다. 그리고 각 그룹 제어 파라미터의 영향을 평가한다.

제어기의 성능평가를 위한 오버슈트의 개선과 빠른 정정시간의 높은 효과를 얻도록 최적화하기 위한 제어시스템의 평가함수는 다음 식(5-2)로 기술되는 하중절대오차적분(ITAE: Integral of time-multiplied absolute value of error)을 이용한다.

$$ITAE = \int_0^{t_j} |e(t)| t dt \qquad (5\text{-}2)$$

그런데, 유전자 알고리즘에 있어서 적합도함수(fitness function)로서는 성능함수의 역수를 취함으로써 식(5-3)의 최대화함수로 변화한다.

$$FitFun = \frac{1}{\int_0^{t_j} |e(t)| t dt} \qquad (5\text{-}3)$$

그리고 선택규칙으로서는 roulette 선택기법을 채용하며, 교차는 K_p,

K_i 각 계수에 대해서 1점 교차(one-point crossover), 즉 각 개체에 대하여 2점에서 교차가 교차확률에 따라 발생되도록 한다.

(4) 비례치 P_c와 P_m에 따라서 선정된 스트링은 교배와 돌연변이이다. 학습과정을 안정화시키기 위하여 최적은 다음 세대의 수로서 유전자 동작없이 남는다. 해에 접근할 때까지 (2)와 (3)을 반복한다. 이러한 과정을 흐름도로 나타내면 그림 4.1과 같다.

이러한 제어방법으로 GA의 최적화 과정은 온라인 추정이며 제어신호는 종래의 PI제어기에 의하여 오프라인으로 최적화한다. 그런 다음, 2개의 PI 파라미터의 적합도는 제어 시스템에 적용한다. 따라서 이 파라미터는 제어목적의 변화에 적응하기 위하여 연속적으로 학습되고 조절된다.

5.2 교류서보전동기의 모델링과 속도제어

본 연구에서는 GA-PI 부가 제어기를 설계하여 K_p, K_I값을 기준으로 안정도를 만족하는 범위 내 에서 교류서보전동기의 속도제어방법을 제안한다.

정밀한 속도 및 위치제어 특성을 얻기 위하여 정확한 기계적 모델링과 관성 모멘트, 부하 등의 시스템 파라미터를 정확히 알아야 한다.

기계적 시스템 모델은 다음 식으로 주어진다.

$$T = J_m \frac{d\omega}{dt} + B_m \omega_m + T_L \qquad (5\text{-}4)$$

여기서 T는 입력토크, J_m은 관성모멘트이고, B_m은 마찰계수, T_L은 부하 토오크이다.

교류서보전동기의 속도제어를 효율적으로 하기 위해서는 빠른 응답 신속한 정정시간과 함께 오버슈트가 없어야 하고 정상상태 오차가 없어야

한다.

이를 위하여 서보전동기의 속도제어에 PI제어, 가변구조제어, 퍼지제어, 적응제어등의 다양한 제어방법이 적용되고 있으나 이중에서 PI제어 방식이 간단한 구조와 구현의 용의성으로 인하여 가장 널리 적용되고 있다. 그림 5.2는 제안된 교류서보전동기의 PI제어시스템의 블록선도이다.

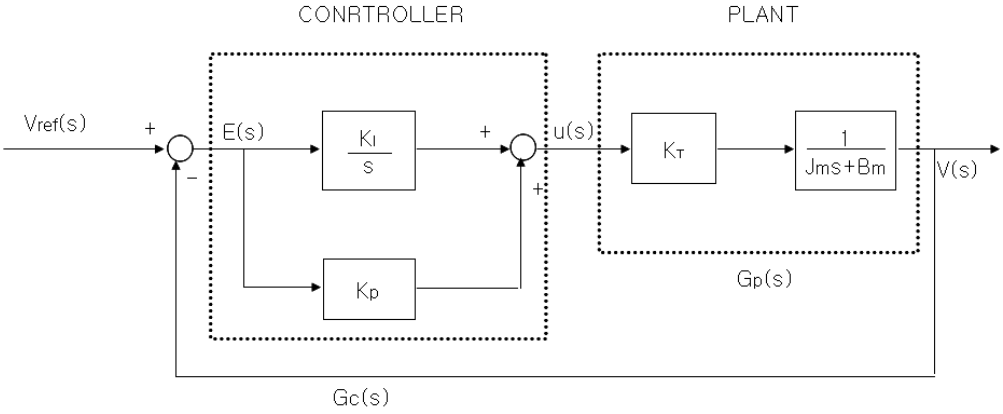


Fig. 5.2 Block diagram of the proposed control system for AC servo motor

입력오차 e(t)이고, 출력 u(t)인 PI제어기는 다음과 같이 주어진다.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt \qquad (5\text{-}5)$$

제어기의 전달함수는

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \qquad (5\text{-}6)$$

이고, 여기서 K_p는 비례이득이며, K_i는 적분이득이다. 이 때 교류서보전동기의 전달함수는 다음과 같이 주어진다.

$$G_p(s) = \frac{K_T}{J_m s + B_m} \tag{5-7}$$

여기서, K_T는 전동기 토오크 상수이다.

그림 5.2의 회전자 속도응답의 전달함수는 다음과 같다.

$$\frac{V(s)}{V_{ref}(s)} = \frac{K_i K_T}{J_m s^2 + (B_m + K_p K_T)s + K_i K_T} \tag{5-8}$$

여기서

$$\zeta = \frac{B_m + K_P K_T}{2(J_m K_i K_T)^{1/2}}, \qquad \omega_n = \left(\frac{K_i K_T}{J_m}\right)^{1/2} \tag{5-9}$$

식(5-9)으로 부터 PI 제어기의 이득은

$$K_p = \frac{(2J_m \omega_m - B_m)}{K_T}, \qquad K_i = \frac{J_m \omega_n^2}{K_T} \tag{5-10}$$

이고, 식(5-8)을 다음과 같은 미분방정식형태로 쓸 수 있다.

$$J_m \ddot{v}(t) + (B_m + K_p K_T)\dot{v}(t) + K_i K_T v(t) = K_i K_T v_{ref}(t) \tag{5-11}$$

이 2차 미분방정식은 다음의 변수를 정의함으로써 다음 방정식으로 쓸 수 있고,

$$x_1(t) = v(t) \tag{5-12}$$

$$x_2(t) = \dot{v}(t) \tag{5-13}$$

결과적으로 상태방정식은 다음과 같다.

$$\dot{x}_1 = x_2 \tag{5-14}$$

$$\dot{x}_2 = -\frac{K_i K_T}{J_m} x_1 - \frac{(B_m + K_P K_T)}{J_m} x_2 + K_i K_T v_{ref}(t) \tag{5-15}$$

벡터형식으로 이 방정식을 쓰면

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_i K_T}{J_m} & -\frac{(B_m + K_P K_T)}{J_m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ K_i K_T \end{bmatrix} v_{ref}(t) \tag{5-16}$$

$$v(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5-17}$$

이다.

Ⅵ. 시뮬레이션 및 실험

6.1 시뮬레이션

GA-PI 제어기에 의한 교류서보전동기의 출력특성 시험 시뮬레이션을 비교 고찰하기 위하여 P제어기, PI제어기 등으로 나뉘어 시뮬레이션 하였다.

GA를 실현함에 있어 필요한 파라미터를 다음 표 6.1과 같이 설정한다.

Table 6.1 Genetic algorithm parameters

개체의 총 길이	44 (22 × 2)
계산종료시 까지의 세대수	200
개체집합의 크기(개체수)	50
교차확률	0.25
돌연변이 확률	0.01

또한 P, I 계수의 탐색영역은 각각 다음과 같다.

$K_P \in [0,\ 50],\ K_I \in [0,\ 10]$

위와 같은 조건하에서 입력을 단위계단함수로 인가했을 때의 시뮬레이션 결과는 다음과 같다. GA-PI 제어기의 우수성을 비교 확인하기 위해 비례제어기 $(K_P = 1)$만을 결합한 경우의 출력특성을 그림 6.1에 나타내었다. 그리고 그림 6.2에 GA에 의해 구해진 최적 P, I로 구성된 제어기를 이용한 경우의 출력특성을 나타내었다. 이 때 구해진 최적계수는 $K_P = 37.8,\ K_i = 0$이다.

그림 6.3에 나타나 있는 과도상태출력과 정상상태출력특성으로부터

GA에 의해 구해진 P, I 계수가 최적치임을 알 수 있으며, 또한 GA-PI 제어기 설계에 대단히 효과적임을 확인할 수 있다.

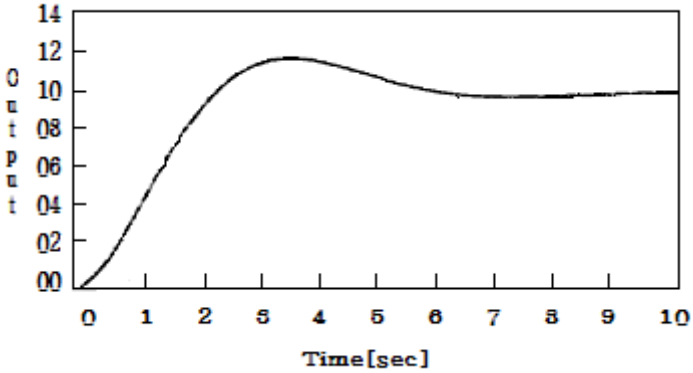


Fig. 6.1 Output characteristics of P controller

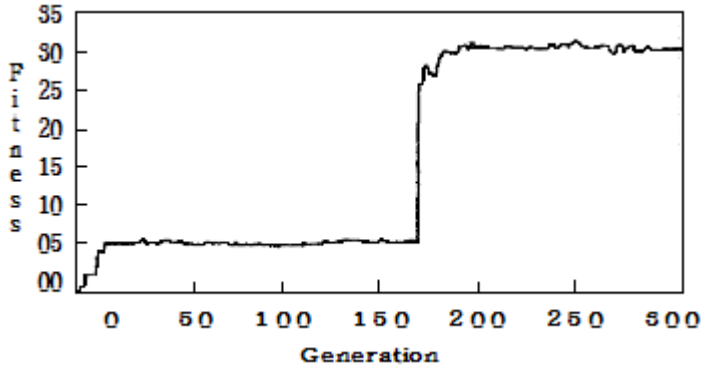


Fig. 6.2 Changes of fitness by evolution

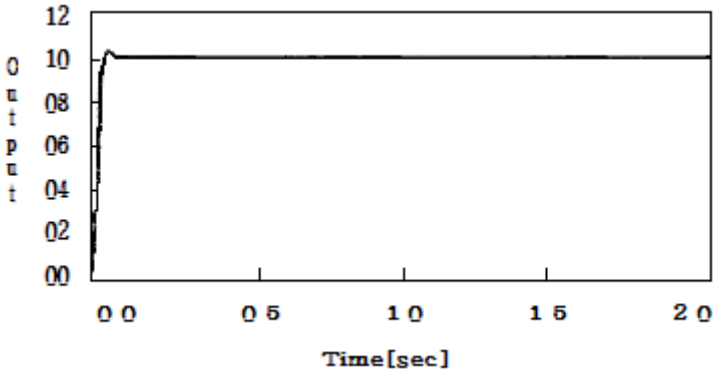


Fig. 6.3 Output characteristics of PI controller

다음은 GA-PI 제어기를 이용한 교류서보전동기 속도실험 컴퓨터 시뮬레이션으로 사용된 서보 전동기는 11[kW] 급으로 정격 속도가 2,000 [rpm]인 전동기의 파라미터는 표 6.2와 같으며 GA-PI 제어기를 사용하였다.

Table 6.2 Parameters of 11[kW]servo motor

정격 전압	220 [V]	회전자 저항 Rr	5. [Ω]
정격 전류	35 [A]	고정자 저항 Rs	4.5 [Ω]
정격 속도	2,000[rpm]	회전자 인덕턴스 Lr	0.243 [H]
고정자인덕턴스 Ls	0.244 [H]	관성 모멘트 Jm	0.082 [Kg · m^2]
상호인덕턴스 Lm	0.239 [H]	마찰계수	0.0035 (kg · m^2/sec]

알고리즘은 컴퓨터 언어 C++를 이용하였으며, 전동기 시스템은 벡터제어 모델에서 5차 비선형 연립 방정식의 형태이므로, 이의 해를 구하기 위해서 Runge-Kutta 방법을 사용하였고 전류제어 주기를 25[ms], 속도제어 주기는 250[ms]로 설정하였다.

제어기와 에뮬레이터의 초기 접속강도는 오프-라인으로 구해지며, 그림 6.4은 GA-PI 제어기를 이용한 경우의 속도응답 파형으로 500rpm, 1500rpm, 0rpm으로 다단계로 속도 지령치를 변화시킨 경우이다. 서보전동기가 PI 제어기에 의해 잘 추종하였고 그림 6.5는 그림 6.4과 동일한 조건으로 하였을 때 PI제어기를 이용한 속도응답 파형으로 매우 잘 추종함을 알 수 있다. 실제 일정 시스템에 맞추어진 PI 제어기는 다른 제어기보다 우수하다. 대부분의 제어기는 그 제어시점에 맞추어진 PI 제어기 보다 양호한 제어가 어렵고 노후화나 공정 및 작업환경 등에서 발생하는 외란 등에 의해 제어환경이 변할 경우 단일 이득을 가진 PI 제어기는 양호한 제어가 곤란하다.

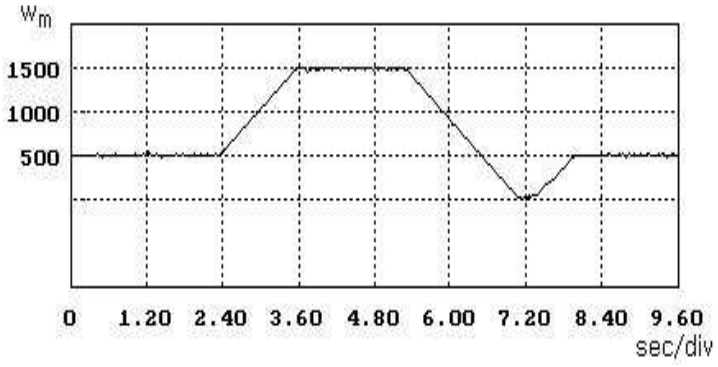


Fig 6.4 Speed characteristics with PI controller using genetic algorithm

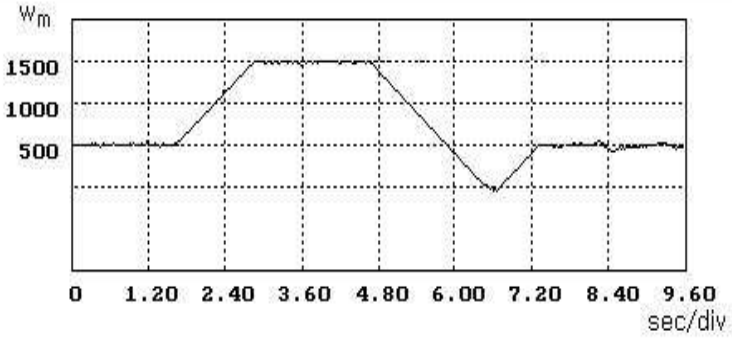


Fig 6.5 Speed characteristics with PI controller

그림 6.5은 PI 속도제어기에 의한 저속 제어로써 매우 양호한 제어를 보이고 있다. 그러나 그림 6.6는 고정자와 회전자의 인덕턴스를 15% 변화시킨 경우로 파형에 변화가 있음을 알 수 있고, 그림 6.7은 GA-PI제어기의 속도제어로써 고정자와 회전자의 인덕턴스를 15% 변화시킴에도 불구하고 미세한 오실레이션만 보이고, 지령속도에 근접한 결과파형을 나

타내고 있다. PI 제어기보다 GA-PI제어기가 환경변화에 강인함을 알 수 있다. 시뮬레이션결과에서 약간의 지연현상이 보이는데 관성계수와 마찰계수가 적당하지 않았을 때, 이러한 현상이 나타났었고 이득값 중에서 K_i의 값이 적당하지 않았을 때도 그러한 현상을 보였다.

그림 6.8은 PI제어에 의한 교류서보전동기의 기동특성 곡선으로 기동시 미세한 오버슈트 현상으로 불안한 기동을 하지만, 추종값에 정확히 도달함을 알 수 있었고 그림 6.9는 GA-PI제어에 의한 기동특성 곡선으로 기동운전시 오버슈트 현상이 거의 나타나지 않는 안정된 출력특성을 나타내고 있다. 그림 6.10은 GA-PI제어기에 의한 교류서보전동기의 속도특성 곡선으로 기동시 오버슈트가 나타나지 않는 안정된 운전특성을 나타내고 있다. 시뮬레이션 결과 일반적인 제어기의 상승시간이 유전자 알고리즘에 의한 PI제어기에 비하여 빠른 반면 오버슈트가 발생하는 것을 알 수 있었으며 제어기의 이득튜닝에 의해서 오버슈트를 제거 할 수 있으나 작업공정이나 주위환경 변화에 따라 이득 계수들이 실시간 자동 튜닝이 어렵고 GA-PI제어기는 상승시간이 느리지만 오버슈트 없이 정상상태에 도달하는 것을 알 수 있었다. 기계가공의 산업용 로봇에 사용되는 서보시스템에서는 오버슈트가 발생하면 공작대상물에 가해질 공구의 회전력이 일정치 못하므로 고전적인 PI 제어에 의한 방법보다는 본 논문에서 제안한 GA-PI제어기가 적합함을 시뮬레이션을 통하여 알 수 있었다.

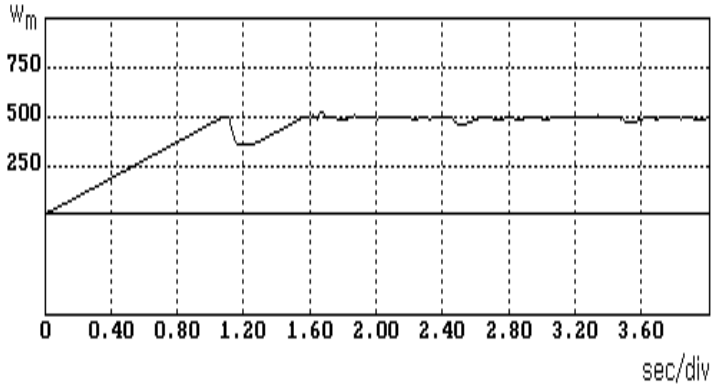


Fig. 6.6 Low-speed characteristics with PI
(Inductance variation 15% of stator and rotor)

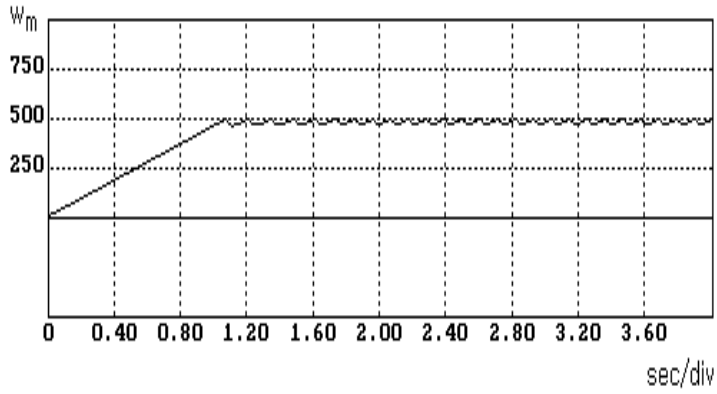


Fig. 6.7 Low-speed characteristics with PI
controller using genetic algorithm
(Inductance variation 15% of stator and rotor)

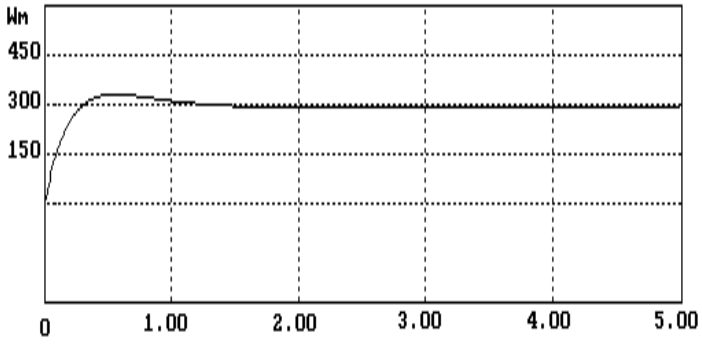


Fig. 6.8 Starting characteristics by PI controller

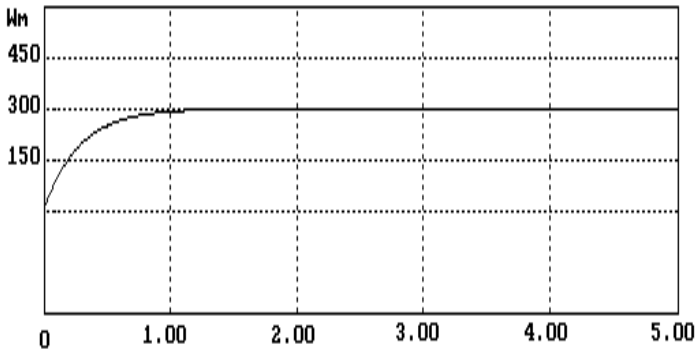


Fig. 6.9 Starting characteristics of PI controller by genetic algorithm

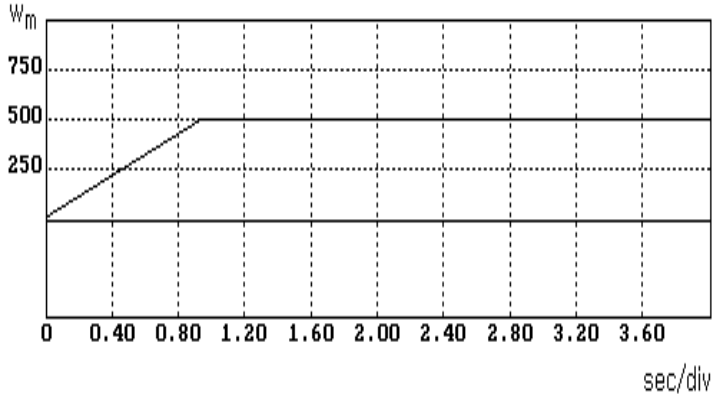


Fig. 6.10 Speed characteristics with PI controller using genetic algorithm

6.2 실험장치의 구성

6.2.1 시스템의 구성

아래 그림 6-11은 서보시스템의 구동 회로이다. 디지털 서보구성은 프로그램에서 주어진 지령정보가 제어기에 의해 처리되고 각 축에 위치지령을 한다. 각 축의 위치 지령정보는 고성능 마이크로프로세서에 전해진다. 마이크로프로세서에는 이러한 정보를 기초로 위치, 속도, 전류의 제어를 일괄 처리하고 PWM제어신호를 출력한다. PWM제어신호는 별도로 장치된 서보 증폭기에 입력되고 서보증폭기는 PWM제어신호를 증폭시켜 교류서보 전동기에 전원을 공급한다.

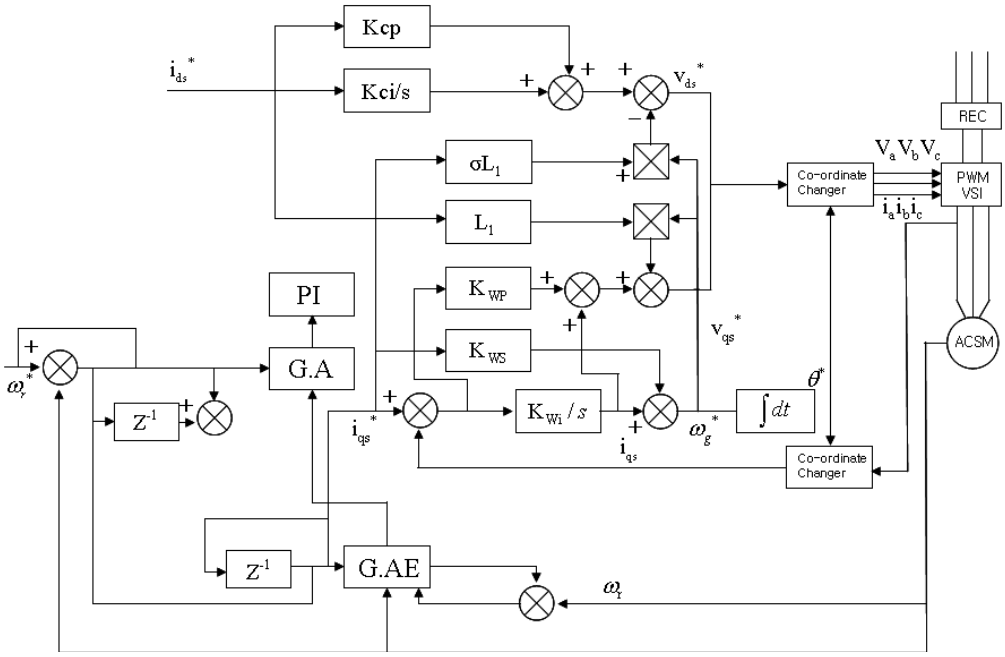


Fig. 6.11 Driving circuit of servo system

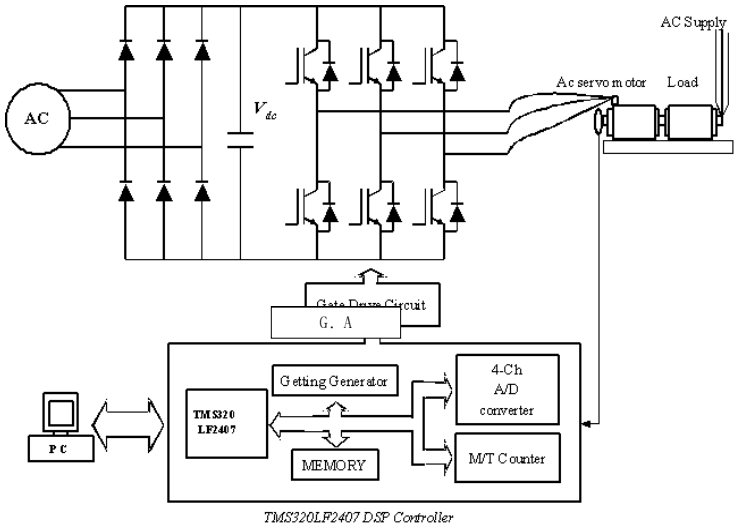


Fig. 6.12 System of AC servo motor

전동기의 구동전류는 서보증폭기를 지나고 전동기의 위치, 속도정보는 펄스 코더 등을 통하여 제어기에 의하여 피드백 된다. 이상과 같이 전동기의 위치, 속도 값 등을 검출하여 추종하여야 조작량을 마이크로프로세서가 선택하여 최종적으로 동작하게 된다.

그림 6-12는 시스템 구성도이다. 제어 대상인 교류서보전동기는 동기전동기이며 부하로는 유도전동기를 사용하였으며, 제어대상 전동기에 전력을 공급하기 위하여 인버터 그리고 제어기를 구성하고 부하에는 3상 변압기에서 전압을 가변하여 입력하였으며 대상 전동기와 반대로 회전하도록 하였다. 인버터 전력용 소자는 고속 스위칭을 얻을 수 있고 PWM으로 인한 전류리플을 줄이기 위해 IGBT를 사용하였다. 속도 및 위치 제어를 위한 GA-PI제어기 알고리즘은 모두 마이크로프로세서에 의해 모두 디지털로 처리하도록 하였으며 33[MHz]로 동작하고 32bit 부동 소수점 연산이 가능한 DSP TMS320LF2407을 사용하였다. 빠른 제어를 위해 서보전용 DSP(Digital Signal Processor)인 TMS320LF2407을 제

어기로 사용함으로써 신뢰성을 높일 수 있을 뿐 아니라 많은 I/O 포트와 주변 장치들을 내장하고 있으므로 다른 제어기를 사용할 때보다 소형, 저가의 시스템을 구성할 수 있었다. 또한, 연산 시간과 제어기의 부담을 줄이기 위해 5,001개의 데이터를 내부 메모리(32K)에 저장해둠으로써 보다 간단한 시스템을 구현하였다.

실험에 사용한 DSP 보드에는 내부의 메모리 외에 16 bit의 EPROM과 16bit의 SRAM을 할당하였다. DSP 내부의 RAM 영역과 외부의 SRAM 영역에 최종의 실행파일을 PC의 에뮬레이터를 통해 다운로딩 하여 모든 제어루틴을 실행한다. 그 외의 보드 구성요소에는 전류의 감지를 위한 3.2μ sec의 변환을 갖는 4개의 A/D 컨버터, 직류전압 검출을 위한 차동 앰프, 그리고 EPLD를 사용하였다. EPLD에는 게이팅 회로, 데드타임 보상, 인터럽트발생, 보호회로, 엔코더 펄스의 체배 등 대부분의 논리회로를 포함하여 프로그램으로 처리하였다. 엔코더는 회전당 펄스수가 200[rpm]를 사용하였다.

교류서보전동기의 제어시스템의 하드웨어는 크게 마이크로프로세서부, 전력변환부, 인터페이스부 등으로 세 부분으로 구성된다. 사용된 DSP는 Texas Instrument사의 TMS320LF2407로 사양은 초고속 연산(30 nsec single instruction cycle)이고 2K × 16비트 고속 RAM, 10bit 16channel의 500ns conversion time을 갖는 ADC, Watchdog timer, CAN(Controller Area Network) Module, SPI(Serial Peripheral Interface) Module, SCI(Serial Communication Interface) Module 등이 내장되어 있다. 전력변환부는 마이크로프로세서의 지시를 받아 3상 교류 220[V]의 상용전원을 전동기의 구동에 적합한 가변주파수 및 가변전압(또는 가변전류)의 교류전원으로 변환하는 부분이다. 전력변환의 정류회로는 3상 다이오드 모듈을 사용하고 인버터는 IGBT를 이용한 3상 브리지 회로를 1개의 소자로 모듈화 하고 게이트 구동회로를 내장하고 있는 IPM(Intelligent Power Module)을 사용하였다.

6.2.2 제어소프트웨어

소프트웨어는 초기화 프로그램과 일정한 주기로 제어 알고리즘을 수행하는 인터럽트 루틴으로 구성된다. 그림 6.13은 제어 소프트웨어의 전체적인 흐름도이다. 초기화프로그램은 초기에 전력변수들과 제어기의 이득값을 지정하고 제어보드에 구성되어 있는 주변소자들의 초기화를 실행한다. 제어를 위한 루틴들은 일정한 주기의 샘플링 시간을 유지해야 하므로 인터럽트 루틴을 구성하였다. 인터럽트는 전류제어를 위해서 타이머 인터럽트(TINTO), M/T를 위해서는 외부 인터럽트(INT2), 속도제어를 위한 외부인터럽트(INT1)로 크게 나눌 수 있다. 전류제어를 위한 샘플링 주기는 DSP의 속도를 고려하여 90[μsec]로 설정하였다. 그리고 속도제어를 위한 샘플링 타임은 전류제어루프 10번마다 한 번씩 하게 된다.

전류제어인터럽트가 발생하면 먼저 인버터를 위한 게이팅이 이루어지고 전류를 읽어 전류제어를 실행한 다음, 공간 전압 벡터 변조 루틴을 실행한다. 실제속도를 계산하기 위한 인터럽트는 M/T 방식으로 속도 측정의 샘플링 주기는 전류제어 샘플링 주기보다 △T 만큼의 시간 후에 처리된다.

속도제어 인터럽트가 발생한 경우에는 먼저 지능제어 알고리즘을 이용하여 속도를 측정하고, 속도제어기에서 전류 지령치를 계산하여 간접 벡터제어를 행한다. 그리고 각 가중치의 초기 값을 실험에 의해 최적 값을 얻을 수 있었고 GA를 이용한 서보 시스템에 의한 교류서보 전동기의 실험결과와 PI제어를 각 단계별로 다양한 실험을 행하였다.

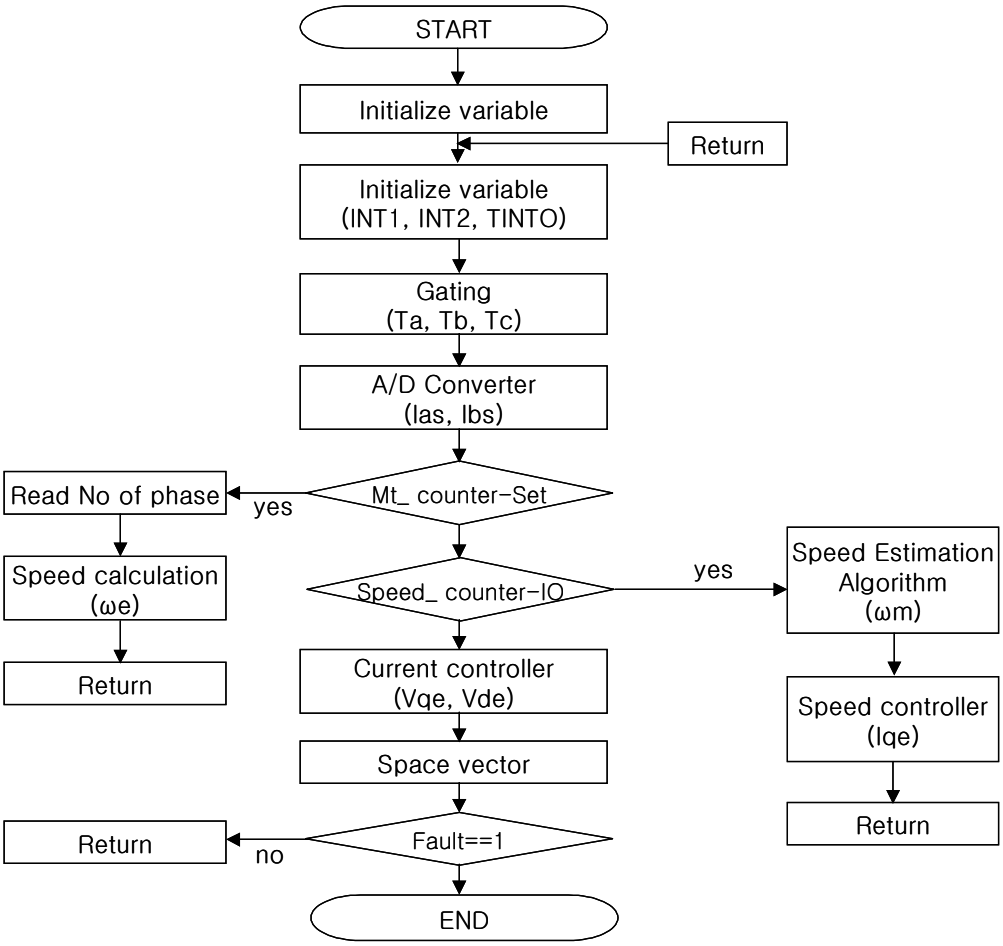


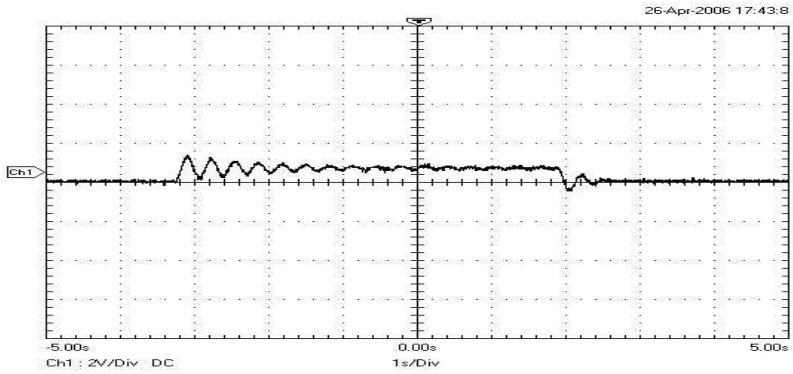
Fig. 6.13 Flow chart of the control software

6.3 실험결과 고찰

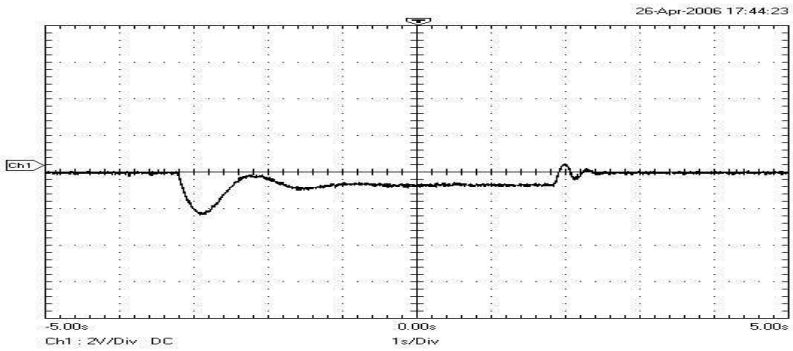
*6.3.1 PI*제어를 이용한 교류서보전동기의 속도지령특성

본 실험에서 PI제어기를 사용하였을 때 실험결과 11kW의 교류서보전동기에서 양질의 응답특성을 보이는 이득 값으로는 K_p=0.75, K_i=0.5 이었다. 본 실험에서의 각각의 실험결과는 PI제어기의 그림 6.14(a)은 정방향회전의 150[rpm]을 추종하는 실험결과를 나타내고 있고, 그림 6.14(b)는 반 시계방향회전의 150[rpm]을 추종하는 결과를 나타내고 있다. 그림 6.14(c)는 시계방향회전의 300[rpm]을 추종하는 실험결과를 나타내고 있고, 그림 6.14(d)는 반 시계방향의 300[rpm]을 추종하는 실험결과를 나타내고 있다. 저속 운전실험에서는 많은 양의 노이즈가 발생되었고, 정지 시에 관성에 의한 응답의 지연현상의 실험결과를 나타내었으며, 미소한 기계적 마찰에 의한 잡음만 관찰될 뿐 희망하는 속도를 정확히 추종하고 있는 것을 보여주고 있다. 그림 6.14(e)는 시계방향, 그림 6.14(f)는 반 시계방향의 600[rpm]을 추종하는 실험결과를 나타내고 있다.

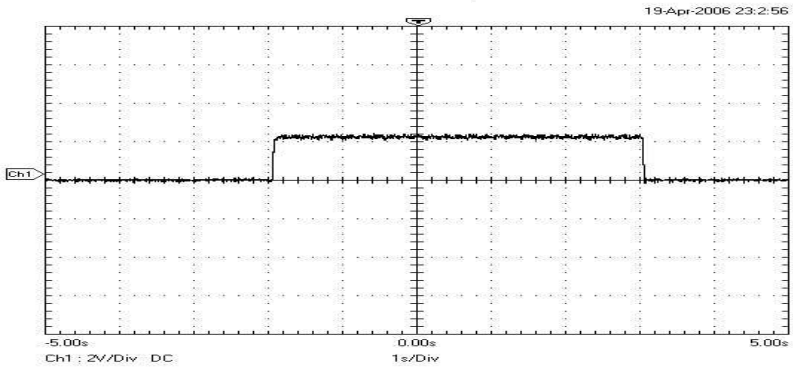
설정 속도 값을 거의 정확히 추종하고 있음을 볼 수 있다. 그림 6.14(h)는 시계방향으로 900[rpm]으로 회전하는 결과를 나타내고 있으며 외란을 인가하지 않았을 때의 결과 파형이다.



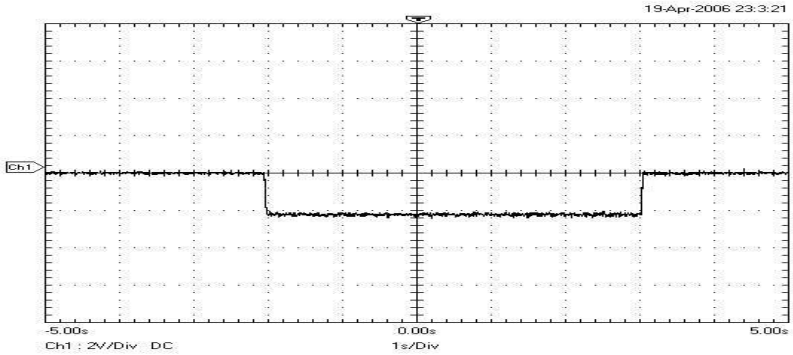
(a) CW, 150[rpm]



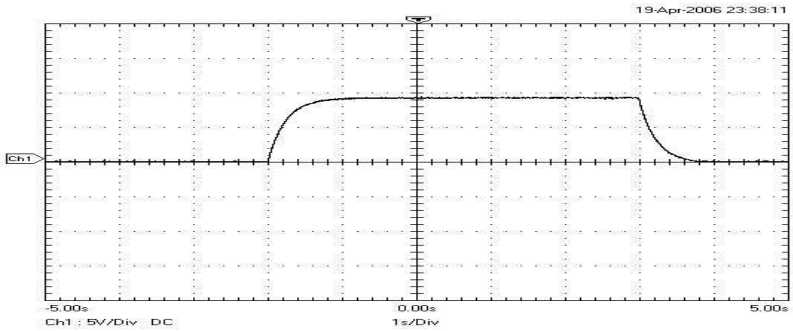
(b) CCW, 150[rpm]



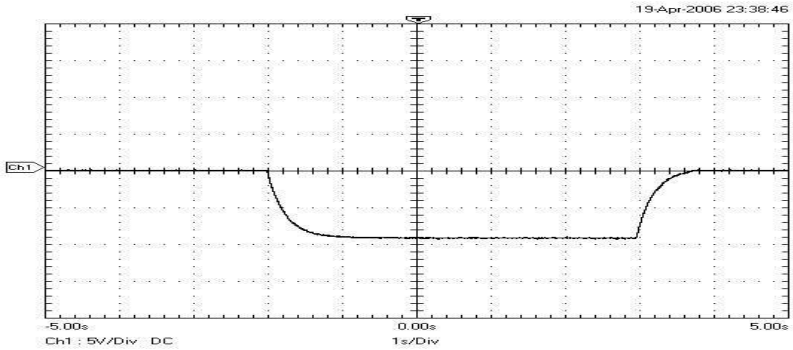
(c) CW, 300[rpm]



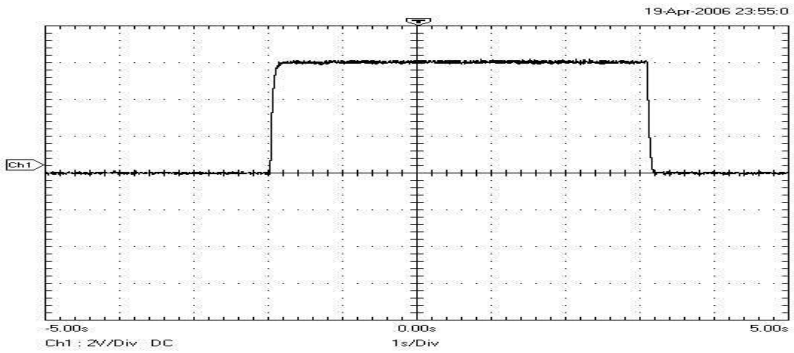
(d) CCW, 300[rpm]



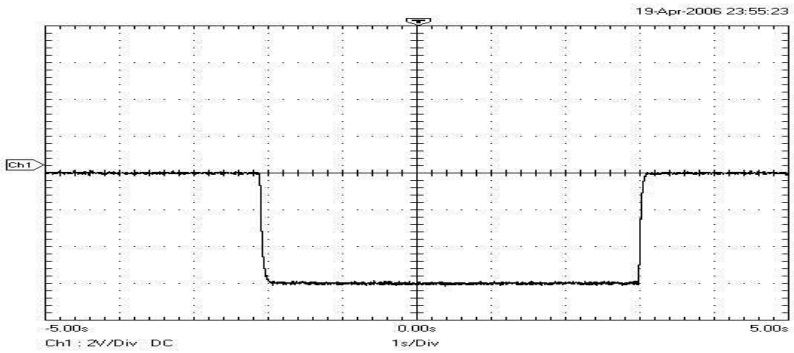
(e) CW, 600[rpm]



(f) CCW, 600[rpm]



(g) CW, 900[rpm]



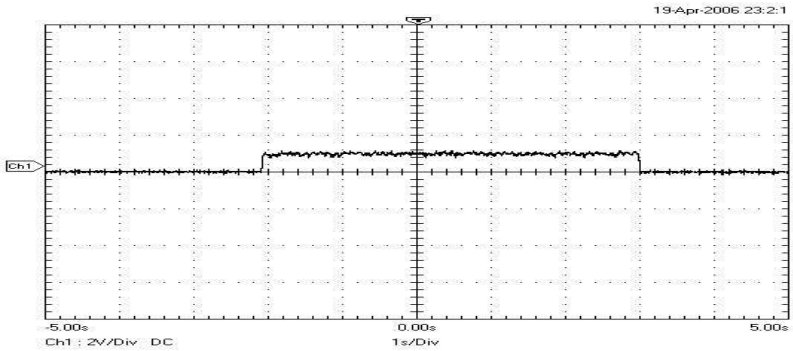
(h) CCW, 900[rpm]

Fig. 6.14 Speed response characteristics using PI controller

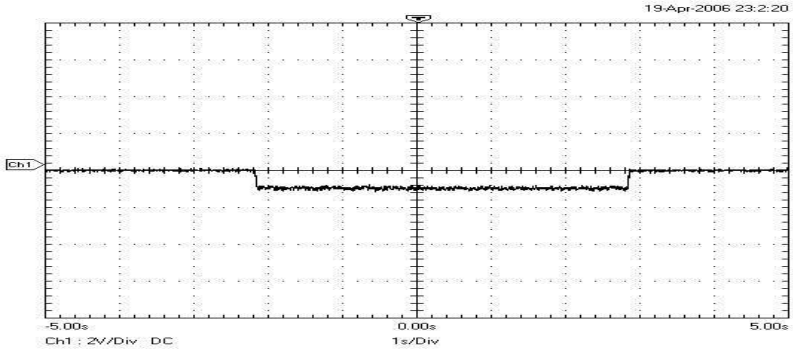
*6.3.2 GA-PI*제어를 이용한 속도지령 특성

GA-PI제어의 실험결과에서 그림 6.15(a)는 정방향으로 150[rpm]회전을 나타내고 그림 6.15(b)는 역방향으로 운전시 150[rpm]회전하였을 때의 결과이다.

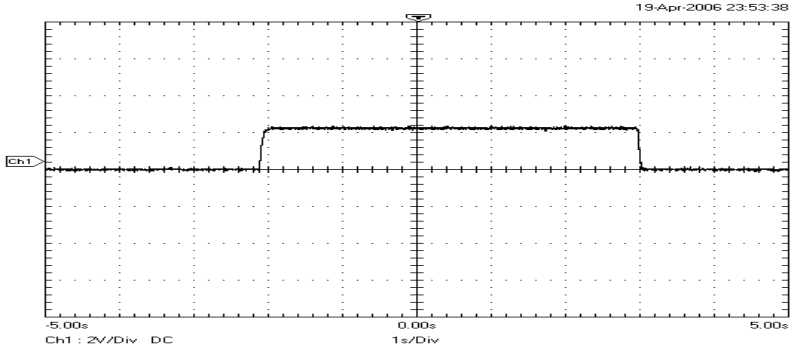
그림 6.15(c)는 정방향으로 300[rpm]회전시켰을 때이고 그림 6.15(d)는 역방향으로 300[rpm]회전시켰을 때의 회전상태인데 미소한 오버슈트만 발견될 뿐 거의 정확한 추종을 하였다. 그림 6.15(e)는 정방 향으로 600[rpm]회전시켰을 때의 결과이고 그림 6.15(f)는 역방향으로 600[rpm]회전 시켰을 때는 정지 시에 미소한 지연현상만 발견되었고, 거의 정확한 추종을 함을 알 수 있다. 그림 6.15(g)는 정방향 으로 900[rpm]회전시켰을 때이고 그림 6.15(h)는 역방향으로 900[rpm]회전하는 실험결과이다. 결과에서 알 수 있듯이 GA-PI제어에서는 미소한 오버슈트 현상만 발생할 뿐 거의 정확한 명령 값을 추종함을 알 수 있었다. 지령치에 거의 정확히 응답을 하였지만 감속운전 때 약간의 오버슈트와 지연이 발생하였을 뿐 설정 값에 정확히 도달하는 양질의 응답특성을 보였다. 속도를 변화시키는 순간에서는 약간의 오버슈트와 지연이 발생되었지만 지령한 값에 거의 정확히 도달함을 알 수 있었다.



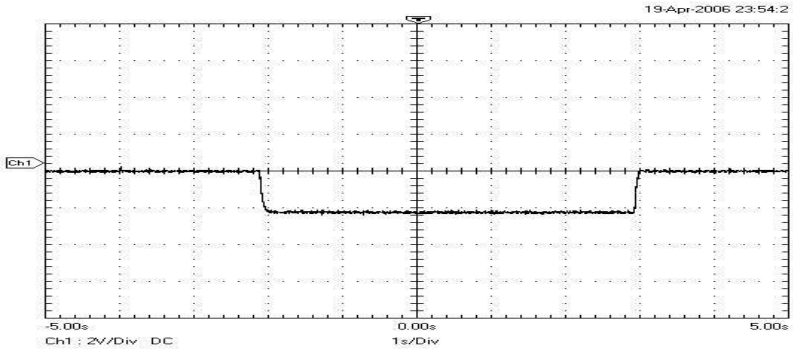
(a) CW, 150[rpm]



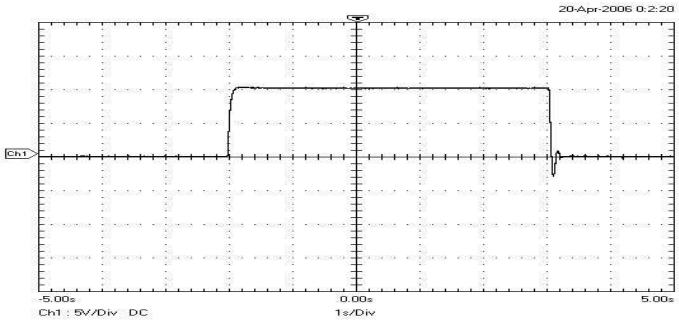
(b) CCW, 150[rpm]



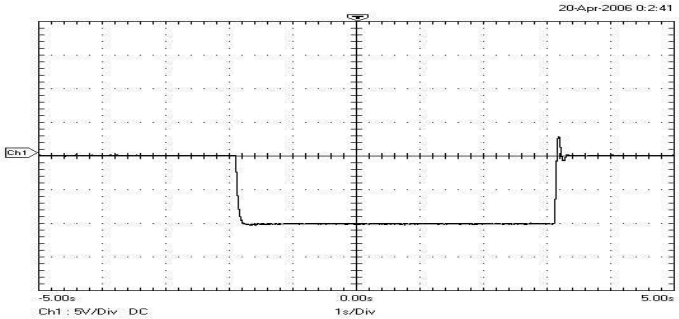
(c) CW, 300[rpm]



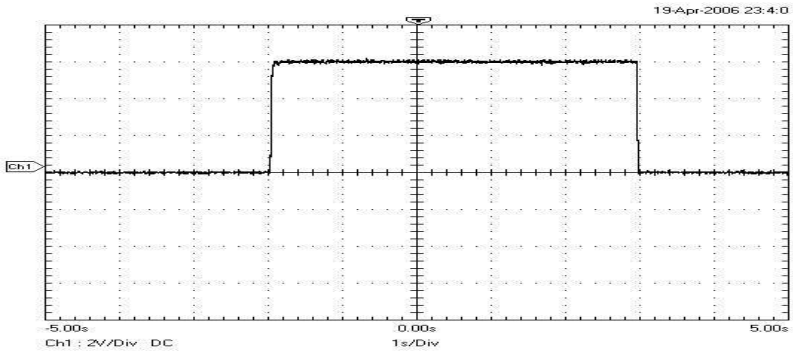
(d) CCW, 300[rpm]



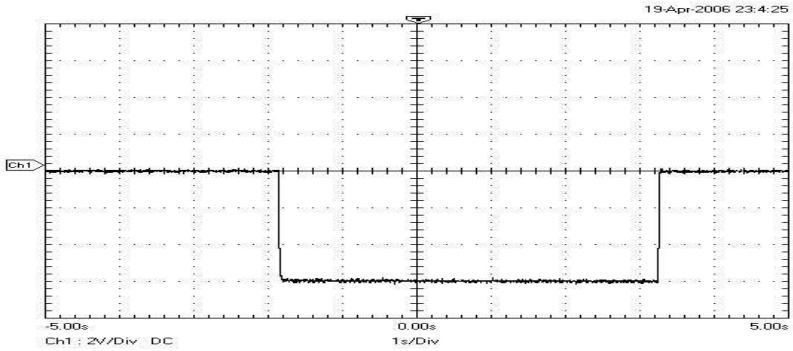
(e) CW, 600[rpm]



(f) CCW, 600[rpm]



(g) CW, 900[rpm]



(h) CCW, 900[rpm]

Fig. 6.15 Speed response characteristics of PI controller using genetic algorithm

Ⅶ. 결 론

본 논문에서는 GA-PI제어기를 설계하고, 제안한 알고리즘에 의해 교류서보전동기의 각 단계별 구간 제어를 구현하였다. 설정된 속도 값을 추종하는 실험을 하여 PI제어, GA-PI제어기 실험의 비교를 통해 GA-PI제어기의 우수성을 확인할 수 있었다.

기존의 제어방법과는 달리 GA-PI제어기를 구성하여 실험에 의해 얻어진 제어이득을 토대로 추종 오차를 검출하여 얻어지는 오차 검출 값을 보정함에 따라 최적의 제어 값을 얻을 수 있으며 다음과 같은 결론을 얻을 수 있었다.

1. 하나의 연산장치 내에서 한 번에 한 가지의 연산만을 순차적으로 수행하는 디지털 컴퓨터의 정보처리방식과는 달리 수많은 뉴런에 분산 저장되는 정보가 동시에 병렬 처리되므로 실시간 처리가 가능하며 몇 개의 공정요소가 고장이 난 경우나 입력정보의 외란이 존재하는 경우에도 전체의 출력에는 영향을 끼치지 않으므로 교류서보전동기의 정밀제어를 위해 시간지연소자를 이용해서 시스템의 현재 입력과 출력뿐만 아니라 과거의 데이터를 입력으로 이용하는 온라인형 GA-PI제어기는 간단한 알고리즘으로 구현하기 쉽고 부하 외란에 강인함을 알 수 있었다.

2. 시스템의 고전적인 식별방법은 정확한 수학적 모델링에 근거하고 있으므로 시스템이 많은 비선형 요소를 포함하고 있는 경우에도 수학적 모델링 없이 비선형함수의 학습능력과 시변 파라미터에 대해 강인한 특성을 가지고 있기 때문에 비선형 시스템 같은 복잡한 시스템을 학습시켜 식별이 가능하게 하였다.

3. GA제어기를 본 논문에서 제안한 알고리즘에 적용하였을 경우 저속

150[rpm]에서 900[rpm]까지 실험한 결과 안정된 출력 특성을 보였고 유전자 알고리즘과 함께 기기 적 상수를 적용한 부하관측기를 부가한 경우 속도추정이 개선됨을 알 수 있었다. GA-PI제어기에 의한 속도 추종 역시 150[rpm] 까지 제어가 가능하고 속도추정의 개선을 위하여 부하관측기가 요구되지만, 응답특성에서 양질의 특성을 얻을 수 있었다.

4. 회전방향을 바꾸는 위치에서 미세한 지연시간이 관측되었는데 이는 회전방향이 바뀌는 데 소요되는 시간으로 본 논문에서 제안한 GA-PI제어가 속도 및 위치 제어에서 지령 값에 근접하게 작동됨을 알 수 있었다.

5. 교류서보전동기의 정밀한 제어실험을 위해서 정방향과 역방향으로 구분하여 단계적으로 구동 실험을 하였고, 실험한 결과 기동 시에는 약 0.03[mm/sec] 내외이며 목표점에 도착 시에는 0.001～0.003[mm/sec]로 측정되었고 또한 본 논문에서 제안한 제어기의 실험 동작 실험의 도착시의 결과를 관찰하면 오버슈트 없이 목표점에 이르는 것을 알 수 있었으며 실정 값과 추종 값을 검출하여 그 오차값 으로 AC서보전동기의 회전속도를 가감함으로써 위치제어에 응용가능성이 있음을 알 수 있었다.

향후 본 연구에서 실험한 결과의 데이터를 다양한 작업환경에서도 적용이 가능한 제어기 개발에 활용하여 본 논문에서 제안한 GA-PI 제어기를 활용한 효율적인 제어를 구현 하고자 하며 본 논문에서 보완되어야 할 부분은 위상제어를 이용한 전동기제어 뿐만 아니라 더욱 정밀 제어가 가능한 로봇의 액추에이터 및 산업용 기계기구 구동원에 쓰일 수 있도록, 본 논문에서 제안한 GA-PI 제어기를 다른 알고리즘으로 복합한 제어기로 개발하여 다양한 산업용 기계기구에 활용하고자 한다.

실험에서 구현한 각 단계의 속도추종실험을 좀 더 세분화하여 저속에서 고속으로의 제어를 구현하면 산업현장의 NC선반이나 머시닝센터의 구동

용으로 응용 가능 할 것 이며 위치제어의 폭을 넓히기 위한 제어방법의 세밀한 연구와 유전자 알고리즘을 좀 더 보완하여 실생활에 활용 가능할 수 있도록 적극적인 연구가 요구된다. 본 논문에서 제안한 GA-PI제어기의 보완을 위해 공작용 기계 등의 각각의 계수를 데이터화 하여 산업용 로봇, 구동용 서보시스템에 적용 할 것으로 기대한다.

참고문헌

1) Mitsuo, Gen, Runwei Cheng, "Genetci Algorithms and Engineering Design", John Weliy & Sons, Inc., 2001.

2) Liu, C. S. and Peng, H., "Disturbance Observer Based AC Servo motor Control", ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 122, pp 332-338, 2000.

3) Takashi Kenjo, "About AC Servo Motor", Clarendon Press, 2001.

4) Bimal K. Bose, "Power Electronics and Variable Frequency AC Servo Motors", IEEE PRESS, 2001.

5) S.K. Park, H.K. Ahn : Controller for AC Servo Motor, ICASF, 2000.

6) U. Itkin, "Control systems of variable structure", JOHNWILLY & ONS, New York, 2003

7) Tracey Lee, T C Chin, D P Mital, Opas Chutatape:A 2-Axis Stepper Motor Motion Control System With Adaptive Capability", IECON'92, Vol.1:442～446, 2000.

8) Torsten soderstrom and Petre Stoica, "System Identification", Prentice Hall, 2000.

9) Kenji lnoue, Junji Yoshitsugu, Shin Shirogane, Prasanna Boyagoda, Mutsuo Nakaoka:"Automatic Learning Control-Based Gain arameter

Auto-Tuning Scheme for AC Servo System", *IEEE,* 2000.

10) Clemens Schaffner, Dierk Schroder, Ulrich Lenz:"Application of Genetic Algorithm to Motor Control", *IPEC'95,* 46~51, 1995.

11) S.Cavalicri:"A Novel Neural Network Model for the Performance Evaluation of Flexible Manufacturing Systems", *IEEE'96,* Vol.3: 1478~1483, 2000.

12) Simon Haykin, "Genetic Algorithms - A Comprehensive Foundation", McMaster University, 2000

13) Guo Qingding, Wang Limei,Luo Ruifu:"Completely Digital AC Servo System Using Neural Network Controller", *ICEPE'95,* 51~55, 2000.

14) Quang P.Ha, M. Negnevistsky:"A AC Servo Motor Drives", IEEE'96, 904~909, 2001.

15) E.Galvan, F.Barrero, M.A.Aguirre, A.Torralba, L.G.Franquelo: "A Robust Speed Control of AC Motor Drives based on Fuzzy Reasoning", IAS'93, PartⅢ:2055~2058, 2003.

16) Bhim.Singh, C.L.Putta Swamy, B.P. Singh, A.Chandra and K.Al-Haddad:"Performance Analysis of Fuzzy Logic Controlled Permanent Magnet Synchronous Motor Drive", IECON'95, Vol.1:339~370, 2000.

17) G. M. Liaw, " A Speed contorller for AC Servo Motor", IEEE Tranceations, Vol. 41, 2001.

18) Vladimir Blasko, "Interaction Between Power Converter and Control-PWM and sampling Issues", 40th IEE Conference on Decison and Control, December 2001.

19) Ying-YuTzou, Tien-Sung Kuo:"Design and Implementation of an FPGA-Based Motor Control IC for Permanent Magnet AC Servo Motors", IECON '97, Vol.2:943~947, 2000.

20) Zelia M. A. Peixoto, Flavia M.f. Sa, Paulo F. Seixas, Benjamin R. Menezes, Porfirio C. Cortizo:"Speed Control of Permanent Magnet Motors using Sliding Mode Observers for Induced e.mf., Position and Speed Estimation", *IECON'95,* Vol.2:1023~1028, 2001.

21) B. Grcar, P.Cafuta, M. Znidaric, D. Dolinar:"Two Stage Nonlinear Control Design of PMAC Serdrive", *IAS,* 3~8, 2000.

22) Wilson, C.E. and Sadler, J.P, Kinematics and Dynamics of Machinery, HarperColiins, New York, 2003.

23) Kenji Inoue, Mutsuo Nakaoka:"Theoretical Considerations of AC Servo Motion Drive Control System with 2-Mass Mechanically Resonant Load", *IECON'96,* Vol.2:1143~1148, 2000.

24) R. Krishnan, R.Monajemy, N. Tripathi:"Neural Control of High

Performace Drives: An Application to the PM synchronous Motor Drive", *IECON'95,* Vol.1:38～43, 2001.

25) Inman, D. J, Engineering Vibration, Prentice-Hall, Englewood Cliffs. NJ, 2000

26) Singh, Heppler, "Shaped input Control of a System with Multiple Modes", Journal of Dynamics Systems, Measurement, and Control Vol. 115, 2000.

27) P. Vas, "Parameter Estimation, Condition Monitoring, and Diagnosus of electrical Machines", Oxford Science Publications, 2001.

28) J R. Cameron, W. T Thomason, A B. Dow, "Detecting Airgap Eccentricity in large Induction Motors", IEE Proc B, Vol. 133, No.3 pp. 155~160, 2000.

29) Dal Y. Ohm, John Mazurkiewicz:"Control of AC Motors for Servo Applications", *PCIM'91*:288～298, 2001.

30) UK-YOUL Huh, Je-Hie Lee:"A Modified Sliding Mode Speed Control Scheme for AC Servo Motor", *IECON'95,* Vol.2:730～735, 2002.

31) S. K. Sung, J. M. Seo, H. J. Nam:"Realtime Distributed Control System for Factory Automation", *ICPE'95,* 830～833, 2000.

32) Uk-Youl Huh, Je-Hie Lee:“A Modified Sliding Mode Speed Control Scheme for AC Servo Motor”, *IEEE,* 730～735, 2001

33) S.A. NASAR, I. BOLDEA, L.E. UNNEWEHR:“ PERMANENT MAGNET, RELUCTANCE, and SELF-SYNCHRONOUS MOTORS, CRC”, 107～137. 2002.

34) Timothy Masters:“About Genetic Algoriths Sourcebook”, John wiley & Sons, INC., 157～220. 2001.

35) F. Ventriglia:“ Genetci Algorithms”, Pergamon Press, 217～250, 2000.

36) Takeshi Yamakawa, " Genetic Algorithm and Its Applications to System Indentification and Prediction of the system Behavior", Proceeding of the 2nd International coference on Logic & GA Networks, pp. 478~482, 2001.

37) Takeshi Yamakawa, " Genetic Algorithm for a control system", 5th IFS World Conference, pp. 1017~1020, 2001.

38) David E. Goldberg, "Genetic Algorithm in search, Optimazation & Machine Learning", Addison wesley, 2000.

39) Simon Haykin, "GA- A Comprehensive Foundation", McMaster University, 2001.

40) R.D. Lorenz. "Tuning of field oriented induction motor controllers for high-performance applications," IEEE, Transactions on Industry Applications, Vol A-10, No.3, pp 393-401, 2000.

41) C. Bohn, D. P. Atherton, "An Analysis Package Comparing PI Anti-Windup Strategies, "IEEE Control Systems, pp. 34-40.2000.

42) TMS320C/F2407 DSP Controller Reference Guide, Texas Instrument, 2000.

43) Texas Instruments, "Implementation of a Speed Field Oriented Control AC Servo Motor using TMS320F240", 2003.

저작물 이용 허락서

학 과	전자공학과	학 번	20047369	과 정	석사, **박사**
성 명	한글: 김 환	한문 : 金 奐		영문 : KIM HWAN	
주 소	경기도 성남시 중원구 금광동 92 단남APT 101-807				
연락처	011-646-4675	E-MAIL	ds4bqx@kdn.com		
논문제목	한글 : 교류서보전동기의 속도제어를 위한 GA-PI 제어기 설계				
	영어 : **Design of a GA-PI Controller for Speed Control of AC Servo Motors**				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집・형식상의 변경을 허락함. 다만, 저작물의 내용변경은 금지함.
3. 배포・전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사 표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 또는 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물의 이용허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송・출력을 허락함.

동의여부 : 동의(◯) 반대( )

2006년 12월 28일

저작자: 김 환 (서명 또는 인)

조선대학교 총장 귀하