DC 모터 서보 제어기의 자동 설계 S/W 개발

The Development of Automatic Design Software for DC Motor Servo Controller

허 경 무, 이 은 오, 조 영 준
(Kyung-Moo Huh, Eun O Lee, and Young June Cho)

Abstract : This paper deals with the development of an automatic design software for DC servo motor control, which provides good performance with rapid response and velocity control accuracy. In the proposed method, the design is automatically executed using Matlab, and iterative learning control algorithms are used in the design process. We applied this method to 50W, 100W, 200W, 300W, 500W, 750W, 1.8kW and 4.5kW DC servo motors which are widely used in the industry. We compare the result of the manual tuning design method with that of the automatic design method presented in this paper. From the experimental results, we can find that the performance of the proposed method is better than that of the manual tuning design method.

Keywords: DC motor, servo control, velocity control, iterative learning control, matlab/simulink

Ⅰ. 서론

DC 모터의 구동 방식은 트랜지스터에 의한 펄스 폭 변조 방식이 주류를 이루는데, 이 방식은 사용 주파수 전원을 정류하여 직류를 얻어 이 직류 전원이 모터에 인가되는 시간 폭을 주파수의 반송파에 의해 변화되어 가변전압을 만들어 모터의 속도 제어를 행한다. 그러나 실제로 모터를 구동시킬 경우, 제어대상인 모터의 정확한 동특성이 파악되지 않으면 성공적인 제어를 하기가 힘들고 따라서 정확하고 빠른 구동이 이루어지기 힘들다. 복잡하면서도 빠른 동특성을 지닌 시스템에 대해 실시간 제어를 하기에는 기존의 적응제어 기법 등이 복잡하고 어려운 것으로 판단되는 경우가 많이 있는데, 이러한 불확실한 동적 시스템을 보다 단순하면서도 정밀하게 제어할 수 있는 방법으로서 학습 제어 방법이 많이 연구되고 있다.

본 논문에서는 제어대상인 모터의 동특성[6]을 고려하여 빠른 응답 특성과 속도 정밀도의 향상을 기할 수 있도록 실시간 제어를 할 수 있는 서보 제어기의 설계에 있어서, Matlab을 이용하여 자동 설계할 수 있는 소프트웨어를 개발하고 그 알고리즘을 제시하였으며, 본 방법을 적용한 실험 결과를 제시한다. 서보 제어기의 핵심인 속도 제어부는 PID 제어 형태로 하되, P,I,D 게인(gain) 각각을 독립적인 반복 학습제어(iterative learning control)에 의해 원하는 성능이 나올 때까지 추적하도록 함으로써, 목표 사양들을 전부 충족시키는 최적의 제어기가 자동으로 설계 되도록 하였다.

특히 모터 설계 시스템과 서보 제어기 설계 시스템은 지금까지 대부분 각각 따로 독립적으로 개발되어 왔지만, 본 논문에서 제시한 자동 설계 방법은 모터 설계 프로그램으로부터 모터가 설계되고[7][8] 모터 파라미터 값이 결정이 되면 그 파라미터 값을 바로 서보제어기 자동 설계 프로그램에 적용하여 서보 제어기 설계가 한꺼번에 이루어지도록 하는 것이 가능하기 때문에, 여러 가지 면에서 진일보한 시도라 할 수 있다. 즉 모터 설계 S/W에 의해 출력된 모터 파라미터들을 본 연구에서 개발된 서보 제어기 설계 S/W에 링크시켜 입력되어 동작시키도록 함으로써, 모터 설계로부터 제어기 설계까지 일원화되어 한꺼번에 설계할 수 있는 통합적인 설계 시스템의 개발이 되도록 하였다.

접수일자 : 2000. 1. 5., 수정완료 : 2000. 5. 6
허경무, 이은오 : 단국대학교 공학부 전자공학전공
조영준 : 생산기술연구원
※ 본 논문은 산업자원부 선도기술개발(G7)사업 지원으로 수행되었음.

Ⅱ. 속도 제어 구조

DC 모터의 속도 제어를 위한 전체 구조도는 그림 1과 같다. 속도 제어부는 과도 특성과 시스템의 부하변동이나 외란에 대한 강인성(robustness)이 우수하면서도 실제 구현하기가 쉬운 일반적인 형태의 PID 제어기로 구성하였다. 그림 1에 전류제어부가 포함된 전체 속도 제어부의 구성을 나타내었다. 이때 전류 제어부는 PI 제어 형태로 하였다.

그림 2는 Matlab을 이용한 실제 시뮬레이션 프로그램으로서, 그림의 하단부에 표시한 바와 같이 모터의 용량과 스위칭 주파수를 설계자가 선택하게 되면, 제어대상 모터

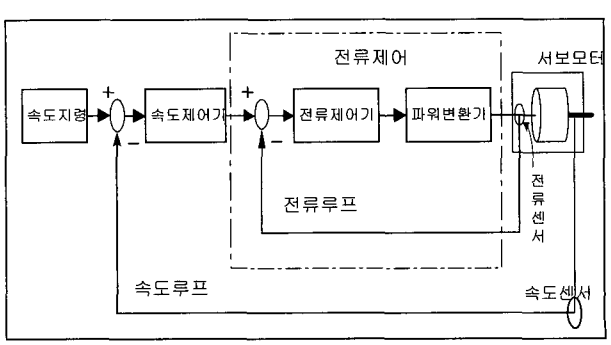
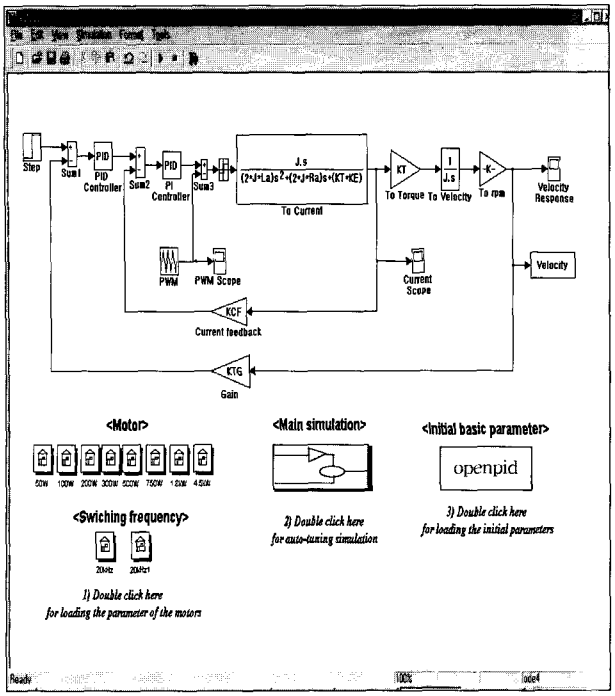


그림 1. 속도 제어 구조도.
Fig. 1. The block diagram of the velocity control.



R_a : Resistance L_a : Inductance

K_T : Torque Const K_E : Voltage Const

K_P : Proportional Gain K_I : Integral Gain

K_D : Derivative Gain K : Unit Conversion Factor

K_{CF} : Current Feedback Gain

K_{TG} : Velocity Feedback Gain

J : Moment of Inertia of Total System ($J = J_M + J_L$)

그림 7. 전류 제어부가 포함된 속도 제어부의 구성도.

Fig. 2. The block diagram of the velocity controller including the current controller.

의 사양이 바로 입력되도록 하는 Matlab상의 블록화 된 메뉴 구성을 하였으며, 이로 인해 편리한 설계가 가능하도록 하였다. 이 블록을 두 번 클릭 하였을 경우 해당 모터의 각 파라미터들(Ra, La, J 등)의 값이 블록도에 있는 해당 변수들로 저장이 되며, 메뉴바에 있는 시뮬레이션을 실행시킬 경우, 수행 후 출력 결과가 스코우프(scope)에 디스플레이 되도록 설계하였다. 중간 수행 결과 데이타를 바탕으로 최종 목표 성능에 접근하도록, 속도 제어기의 PID 게인을 갱신하여 시뮬레이션에 적용시켜 수행시키며, 이러한 과정을 자동적으로 계속 반복하여 최종 목표 성능에 근사한 결과 즉, 최종 목표 사양을 만족하는 값으로 접근하도록 계속 시뮬레이션 하도록 한다. 목표 성능이 만족되면 수행을 마치게 되며, 이 때의 게인 값들이 최종 설계값이 된다.

Ⅲ. 속도 제어기의 자동 설계 프로그램

본 논문에서 설계하고자 하는 DC 모터 속도 제어기의 목표 성능 사양은 통상적인 자동화 장비용 서보 모터 사양인 표 1과 같이 하였다.

표 1. 목표 성능 사양.

Table 1. Desired performance specification.

• 정격속도에서의 속도 변동율(steady-state error) : ± 1rpm
• 정격속도에서의 overshoot : 5% 이내
• 정격속도에서의 settling time(Ts) : 10 msec 이내 (단, 4.5KW 모터의 경우에는 20msec 이내)
• 적용가능한 부하 회전관성 : 모터 회전관성의 5배까지

여기에서 일반적으로 정착시간(settling time)이라 함은 정상상태(3000/2000rpm의 정격속도)의 2% 범위 내에 정착하는 시간을 말하나[6], 본 논문에서의 정착시간은 정밀도를 고려하여 정상상태의 기준속도(3000/2000rpm)와 실제 출력속도의 오차가 목표 오차(±1rpm) 범위 내로 정착하는데 걸리는 시간으로 정의하였다.

본 자동 설계 프로그램에서는 제어용으로 많이 쓰이는 Matlab/Simulink를 이용하여 설계하였으며, 그 설계 과정은 다음 흐름도에 나타내었다.

먼저 프로그램이 실행이 되면, 그림 2와 같은 자동 설계 기본 블록도가 화면에 나타나며, 이 때 흐름도의 두 번째 과정에서 PID 게인의 초기 값과 프로그램 수행시 필요한 파라미터값을 설정하며, 게인의 초기 값은 수동 시뮬레이션을 통하여 얻은 작은 용량에서 큰 용량으로 게인을 확장하기 편한 P 게인이 100, I게인이 1, D 게인이 0.01로 설정하였다. 세 번째 과정에서 제어하고자 하는 대상 모터의 파라미터들을 입력한다. 입력 대상 항목은 표 2에 나타내었으며, 입력방법에는 직접 모터 파라미터들을 대화 상자에서 입력하는 방식과 Matlab Simulink에서 직접 입력하는 방법 두 가지 모두 가능하다. 만약 모든 파라미터 값들이 설정이 안 되었을 경우 다시 입력을 요구하게 되며, 위의 과정을 걸치게 되면, 자동 시뮬레이션하기 위한 기본 초기화 단계가 마무리 된 것이다. 이 과정까지가 데이터 설정 단계이다. 네 번째로 속도 제어부의 PID 게인을 자동 설계하는 단계로 구성하였다. 그림 4는 자동 설계을 위한 대화상자이며, Matlab GUI를 이용하여 구성하였다.

그림 4의 대화상자에서 'Simulation Start'을 실행하여 시뮬레이션을 수행하게 되고, 또한 진행되는 결과 값들은 'Simulation result display' 부분에 표시하여 관찰할 수 있고, 속도 파형은 대화상자 상에서 직접 모니터링 할 수도

표 2. 모터 파라미터 입력항목.

Table 2. The input item of the motor parameters.

모 터 파 라 미 터 입 력 항 목			
항 목(기호)	단위	항 목(기호)	단위
전기자권선저항(Ra)	Ω	회전자 관성(J)	$kgcms^2$
전기자인덕턴스(La)	H	Current Feedback Gain(K_{CF})	V/A
토오크정수(K_T)	kgcm/A	최대속도(ω_{MAX})	rpm
유기전압정수(K_E)	V/rpm	Velocity Feedback Gain(K_{TG})	V/rpm

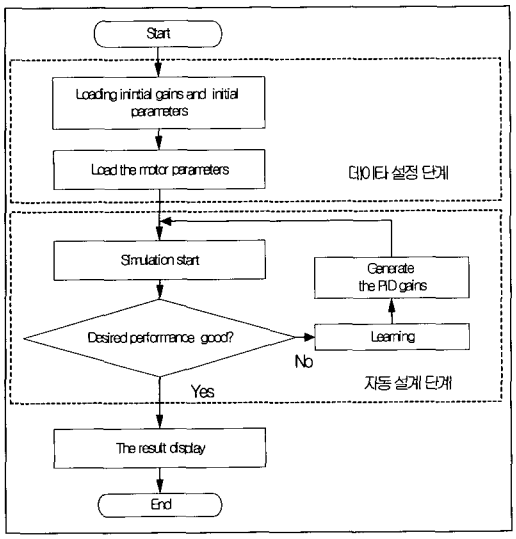


그림 3. 자동설계 프로그램 수행과정 흐름도.

Fig 3. The flow chart of the automatic design program accomplishment.

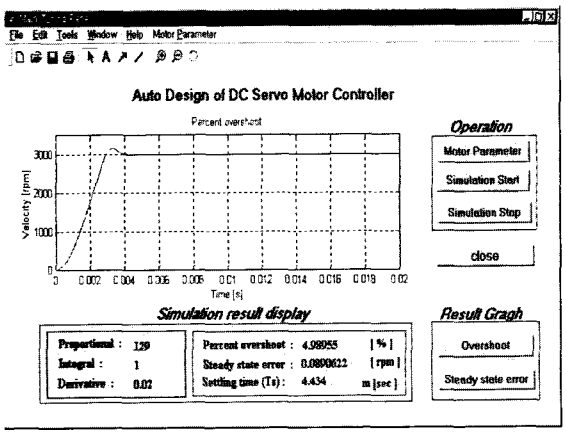


그림 4. 자동 설계 대화 상자.

Fig. 4. The automatic design dialog box.

있다. 또한 결과 값들은 또한 저장되어지도록 설계되었다. 그리고 'Overshoot'메뉴와 'S-S Error'메뉴를 첨가하여 목표 성능과 자세히 비교할 수 있도록 하였다.

목표 성능에 근접한 결과를 얻기 위해서 수동 설계시에는 스코우프 상에 나오는 시뮬레이션 결과를 설계자가 직접 판단하여 목표 성능에 근접할 수 있도록 PID 게인을 재조정하여 입력한 후 다시 시뮬레이션을 하여 그 결과를 관찰하는 방법을 반복하였다. 이는 시간과 시뮬레이션 반복 횟수에서 많은 낭비 요인이었지만, 자동 설계 프로그램에서는 이러한 모든 과정을 프로그램 스스로 자동으로 판단하여 수행하도록 알고리즘을 구현하였다.

Ⅳ. 자동 설계 알고리즘

속도 제어기에서의 제어 방식은 과도 특성이 우수하고, 또한 시스템 파라미터의 변동이나 외란에 대한 강인성(robustness)이 우수하면서도 실제 구현하기가 쉬운 일반적인 형태의 PID 제어 방식을 채택하였다.

그림 3에서의 자동 설계 단계에서 PID 게인을 결정하며, PID 게인 결정 과정은 다음과 같이 하였다.

첫째 초기 PID 게인 값에 의해 실행된 결과를 분석하여, 시스템 특성을 파악하여 부족제동(underdamping)인지 아니면 과제동(overdamping)인지를 판단한 다음, PID 게인 중에 어느 것을 먼저 학습(learning) 할 것인지를 판단하여, 각각의 조건에 맞는 상황의 게인을 갱신해 나간다. 만약 시뮬레이션 결과가 부족제동일 경우, 최소 rpm(최대 rpm 다음에 계산된 최소 rpm)이 정격 속도(3000rpm or 2000rpm)보다 작을 시, 정격속도에서 최소속도의 차를 오차를 갖는 D 게인 학습에 들어간다. 이는 그림 5에 나타나 있는 'PID control I'의 과정이다. D 게인 학습의 결과, 과제동으로 판단이 된 경우에는 P 게인을 증가시켜 반응 속도를 빠르게 한다. 이런 과정을 반복하여 최소속도가 정격속도에 미치는 동시에 overshoot가 5% 이내로 판단(Overshoot 판단(I))되면 이 때부터 settling time과 steady state error를 목표 사양과 비교하여 비교 결과를 가지고 각 게인들의 학습에 들어간다.

둘째 판단 기준은 정착시간(settling time)이 목표 성능에 도달하는 가이다. 정착시간의 기준은 정격속도±1rpm에 도달하는 시간이며 이 때의 시간은 10ms 이내이어야 한다. 시뮬레이션 결과로 나타난 정착시간과 기준 정착시간의 오차가 P 게인 학습 제어 계산식의 오차항의 대상이 된다. 즉 10ms가 목표 사양이며 이와 시뮬레이션에서 얻어진 결과의 차가 P 학습제어에 적용이 된다. 만약 시뮬레이션에서 얻어진 settling time이 10ms내에 이르렀을 시에 settling time을 저장한 후 한번 더 P 게인 학습을 한다. 학습 결과 settling time이 저장된 settling time보다 더 빠를 경우 계속 P 게인 학습을 반복 수행한다. 최종 시뮬레이션 후 얻어진 settling time이 저장된 settling time보다 크게 나타나면 바로 그림 5에 있는 'PID control Ⅱ'과정을 마친다.

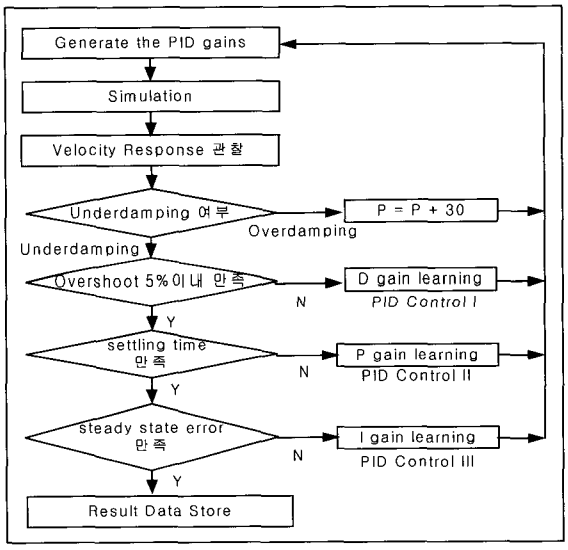


그림 5. 반복 학습 수행과정 블록도.

Fig 5. The flow chart of the iterative learning performance process.

세 번째 판단 기준은 steady state error가 목표 성능에 도달하는 가이다. steady state error를 판단 대상으로 하여 I 게인 학습을 적용하였다. 판단 방법은 위의 P 게인 학습 방법과 동일한 방법으로 기준 steady state error(±1rpm)과 시뮬레이션에서 얻어진 steady state error의 차를 I 게인 학습 제어의 오차의 대상으로 하였다. 즉, ±1rpm이 목표 사양이며 이와 시뮬레이션에서 얻어진 steady state error의 차가 I 학습제어 오차에 적용이 된다. 시뮬레이션에서 얻어진 steady state error가 ±1rpm 내에 들지 못했을 시에는 I 게인 학습을 반복 수행한다. 최종 steady state error가 목표 사양에 만족이 되면 그림 5에 있는 'PID control Ⅲ' 과정을 마친다.

'PID control Ⅱ'와 'PID control Ⅲ' 과정을 마쳤다면 그림 5에서 보는 대로 판단(Ⅰ), 판단(Ⅱ), 그리고 판단(Ⅲ) 조건을 만족하는지를 판단하여 위의 3 판단 조건들을 모두 만족시킨다면 시뮬레이션을 멈추고 그 결과값을 모니터상에 디스플레이 시킨다.

위의 학습제어 수행 과정은 그림 5의 흐름도에 나타내었다. 위의 수행 과정에 적용한 학습제어 방법은 반복적인 온라인 제어기법으로서 계산량이 적고 대상 시스템의 동특성에 대한 사전 정보가 많이 요구되지 않는 장점을 지니고 있다.

Arimoto는 [1]에서 다음의 간단한 학습 제어 알고리즘을 제안하였고,

$$u_{k+1}(t) = u_k(t) + \Gamma \frac{d}{dt} e_k(t) \quad (1)$$

우리는 이미 [3]에서 학습제어의 수렴 성능을 향상시킬 수 있고, 시스템의 강인성과 안정성이 향상될 수 있는 방법으로서, 비선형 시스템에도 적용이 가능한 피드백 사용형 2차 반복 학습제어 알고리즘((2))을 제안한 바 있다.

$$u_{k+1} = P_1 u_k + P_2 u_{k-1} + Q_1 e_k + Q_2 e_{k-1} + R e_{k+1} \quad (2)$$

본 연구에서는 목표 사양들을 만족시킬 수 있는 제어기의 각 PID 게인들은 학습 과정을 통하여 PID 게인들의 값들을 선정하였으며, 학습 알고리즘으로는 (2)의 가장 간단한 형태인 (3)과 같은 알고리즘을 적용하였다.

$$u_{k+1} = u_k + Q e_k \quad (3)$$

먼저 그림 7의 'PID control Ⅰ'에서는 D 게인을 (4)의 학습 알고리즘에 의해 구하였다.

$$K_{D,k+1} = K_{D,k} + Q ve_k \quad (4)$$

여기에서 ve_k는 목표속도와 k번째 반복 수행 시 오버슈트 발생 후의 최소속도와의 차이(단위 : rpm)이고, Q는 0.00025로 설정하였다. 이 때의 Q 값은 실험에 의해 선정된 값이다. 다음에 'PID control Ⅱ'에서 P 게인을 (5)의 학습 알고리즘에 의해 구현하였다.

$$K_{P,k+1} = K_{P,k} + Q ste_k \quad (5)$$

이때의 ste_k는 목표 settling time(ST)과 k번째 반복수행 시의 실제 ST와의 차이(단위 : ms)이며, Q 역시 실험에 의해 1.5로 선정하였다. 끝으로 'PID control Ⅲ'에서는 I 게인을 (6)의 학습 알고리즘에 의해 구하였다.

$$K_{I,k+1} = K_{I,k} + Q se_k \quad (6)$$

여기에서 se_k는 목표 steady-state error(SSE)와 k번째 반복 수행시의 실제 SSE와의 차이(단위 : rpm)를 나타내며, 이때의 Q값도 실험에 의해 0.05로 선정하였다.

이들 일련의 학습과정을 통하여 최종적으로는 모든 목표 사양을 충족시킬 수 있는 서보 제어기를 자동 설계할 수 있도록 하였다. 이와 같이 프로그램 스스로 판단할 수 있는 알고리즘을 개발하여 프로그래밍 함으로써, 계산되어져 나오는 성능 결과를 스스로 판단하여 성능을 개선시키는 방향으로 PID 게인을 조정하는 방법을 반복함으로써 최적의 PID 게인을 자동 결정한 후, 다이얼로그 박스 안의 하단에 있는 편집 박스에 결정된 PID 값을 표시하도록 하였다. 이상의 학습 과정을 통해 원하는 성능을 만족시키는 결과가 나올 때에는 자동적으로 최종 결과를 표시하고, 'Controller Design Completed'가 화면에 나타나게 된다. 이때 편집 박스에 보여주는 PID 게인 값이 최종적인 PID 게인 값이 된다.

이상에서 설명한 다이얼로그 박스의 형태는 앞의 그림 3과 같다. 수동 설계에서 사용자가 일일이 수행하고 판단하였던 이 모든 과정들이 자동 설계 프로그램에서는 그림 3의 대화상자 안에서 자동 실행되어진다.

Ⅴ. 시뮬레이션 결과 비교 및 고찰

시뮬레이션 대상 모터로서는 자동화 장비에 많이 쓰이는 50W, 100W, 200W, 300W, 500W, 750W, 1.8kW, 4.5kW 급의 DC 모터를 선정하였으며, 각 모터의 파라미터들은 표 3과 같다. 이 모터들 각각에 대하여 표 1의 제어 목표가

표 3. 모터의 사양.
Table 3. The motors specification.

모터별 사양									
항목		사양							
항목 (기호)	단위	50W	100W	200W	300W	500W	750W	1.8kW	4.5kW
전기자 권선저항 (Ra)	Ω	24.03	20.53	5.10	2.80	1.40	1.15	0.56	0.10
전기자 인덕턴스 (La)	H	18.1×10^{-3}	20.8×10^{-3}	9.8×10^{-3}	5.6×10^{-3}	5.41×10^{-3}	5.3×10^{-3}	4.35×10^{-3}	1.80×10^{-3}
토오크 정수 (KT)	kg cm/A	2.30	2.83	2.85	2.75	3.03	4.4	7.75	7.85
유기전압 정수 (KE)	V/rpm	23.6×10^{-3}	29.1×10^{-3}	29.2×10^{-3}	28.2×10^{-3}	31.1×10^{-3}	45.0×10^{-3}	79.6×10^{-3}	80.6×10^{-3}
회전자 관성 (J)	kg_{cms^2}	0.078×10^{-3}	0.128×10^{-3}	0.517×10^{-3}	0.843×10^{-3}	2.775×10^{-3}	3.80×10^{-3}	9.8×10^{-3}	90.0×10^{-3}
Current Feedback Gain (K_{CF})	V/A	6.8/3	6.8/4	6.8/8	6.8/11	6.8/16	6.8/16.5	6.8/34.0	6.8/77.8
최대속도 (ω_{MAX})	rpm	4,500						3,000	
정격속도 (ω)	rpm	3,000						2,000	
지령신호 (V_{rd})	V	± 6						± 6	
Velocity Feedback Gain (K_{TG})	V/rpm	6/3000						6/2000	

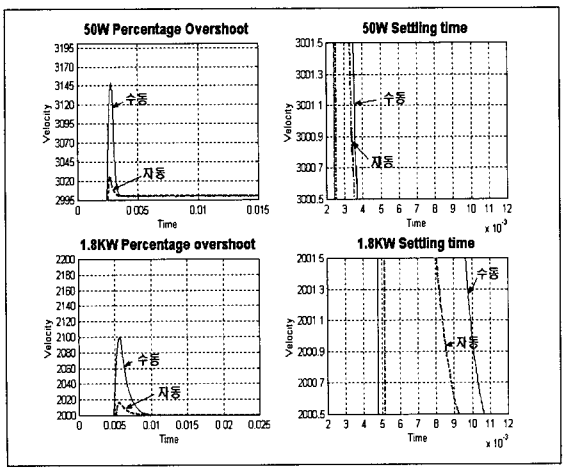


그림 6. 시뮬레이션 결과 그래프(수동, 자동).

Fig. 6. The simulation result graph(Manual via Auto).

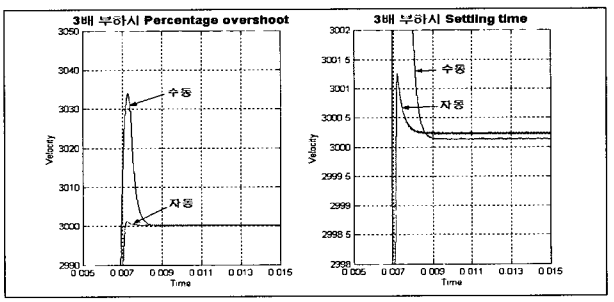
표 4. 제어 게인 및 제어 성능 결과.

Table 4. The control gains and the simulated result of gain tuning.

모 터	Control Gain						지령속도 (3000/2000[rpm])			
Watt [W]	P		I		D		P.O (%)		Ts (ms)	
	수동	자동	수동	자동	수동	자동	수동	자동	수동	자동
50W	300	68.03	1	0.70	0.036	0.01	4.8	0.73	4	3.35
100W	310	78.93	1	0.95	0.054	0.01	5	2.89	5	4.47
200W	335	81.06	1	0.90	0.12	0.03	4.9	0.94	6.4	5.78
300w	353	112.79	1	1.10	0.13	0.045	4.8	0.87	6.3	5.70
500W	900	71.97	1	0.07	0.65	0.042	4.8	1.18	10.6	9.98
750W	603	90.33	1	0.02	0.47	0.064	4.9	1.57	10.4	9.99
1.8kW	700	126.85	1	1.10	0.64	0.12	4.9	0.88	10.6	8.48
4.5kW	704	79.78	1	0.95	1.29	0.12	4.8	3.17	20.2	19.46

이루어질 수 있도록 속도 제어부를 설계하였다. 수동으로 설계한 제어기의 성능 결과와 자동 설계 방법에 의해 설계한 제어기의 성능결과는 표 5와 같다. 여기에서 수동 설계란 속도 제어부의 P,I,D 게인을 일일이 수동으로 변화시켜가며 목표 성능을 만족시키는 방향으로 조정해나가는 방법으로서, 게인을 찾는데 많은 시간을 요하고 최적의 결과를 얻어내기가 상당히 어렵다. 그림 6은 각 7개의 모터 중 적은 용량의 모터(50W)와 큰 용량의 모터(1.8kW)를 선별하였으며, 선별된 모터에 대한 수동 설계 시 얻은 최상의 결과와 자동 설계 시 얻은 결과를 그림으로 나타낸 그림이다. 그래프에서 알 수 있듯이 적은 용량의 모터에서 뿐 만 아니라, 큰 용량의 모터에서도 percentage overshoot(P.O)와 settling time(T_s)에서 좋은 결과를 얻을 수 있었다.

표 4의 결과에서 알 수 있듯이, 모든 용량의 모터에 대해 자동 설계에 의해 구한 결과가 수동 설계에 의한 결과보다 오버슈트가 훨씬 더 작고, settling time도 더 짧아진 것을 알 수 있다. 여기에서 정격속도에서의 속도 변동율은 모두 ±1rpm 이내로 잡힌 상태이다. 또한 수동일 때의 각 P 게인과 D 게인들은 큰 값을 지닌 반면 자동 설계에서 얻어진 게인들의 값들은 상대적으로 작아졌기 때문에 게인 마진과 위상 마진에 있어서 그만큼 여유가 생겼음을 알 수 있다. 그리고 무엇보다도 제어기 설계를 자동화함으로 인해, 짧은 시간 내에 좋은 성능의 서보 제어기를 설계할 수 있다는 것을 확인할 수 있다. 그리고, 그림 7은 100W의 모터를 대상으로 모터 회전관성의 3배의 부하가 걸렸을 때의 성능 결과이며, 이 결과에서 알 수 있듯이 수동에서의 결과 보다 자동에서의 결과가 응답속도가 더 빠르고, 정착시간(settling time)도 짧아졌음을 알 수 있다.



[수동(P:310 I:1 D:0.054) 자동(P:78.93 I:0.95 D:0.01)]

[Manual(P:310 I:1 D:0.054) Auto(P:78.93 I:0.95 D:0.01)]

그림 7. 시뮬레이션 결과(3배 부하가 있을 경우 (100W)).

Fig. 7 The simulation result(the case of being the three times load(100W)).

VI. 결론

본 논문에서는 제어대상인 모터의 동특성을 고려하여 빠른 응답 특성과 속도 정밀도의 향상을 기할 수 있도록 실시간 제어를 할 수 있는 서보 제어기의 설계에 있어서, Matlab을 이용하여 자동 설계할 수 있는 소프트웨어를 개발하고 그 알고리즘을 제시하였으며, 본 방법을 적용한 시뮬레이션 결과를 제시하였다. 이 결과로부터, 제시한 자동 설계 방법에 의한 속도제어 성능이 매우 뛰어남을 알 수 있었다. 본 연구에서 제시한 서보 제어기 자동 설계 S/W는 모터 설계 시스템과의 통합이 바로 가능하므로, 모터 설계로부터 제어기 설계까지 한꺼번에 설계할 수 있는 통합적인 설계 시스템의 구현이 가능하다는 장점을 갖고 있다.

참고문헌

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning", *J. of Robotic Systems*, vol. 1, no. 2, pp. 123-140, 1984.

[2] Gu, Y. and Loh, N.,"Learning control in robotic systems", *Proc. IEEE Int'l Symposium on Intel-ligent Control*, Philadelphia, Pa. pp. 360-364, 1987.

[3] 허경무, 우광준, "비선형 시스템에 적용 가능한 피드백 사용형 2차 반복 학습제어 알고리즘", 제어·자동화·시스템공학 논문지, 제4권, 제5호, 1998년, 10월, pp. 608-615

[4] Kyungmoo Huh and Hankyung Bae, "A design of AC servo motor drive system -BLDC type-,"

IEEE International Symposium on Industrial Electronics, pp. 658-662, Budapest, Hungary, Jun. 1993.
[5] K. Àstrôm and T. Hâggludn, "PID controllers: Theory, design, and tuning"
[6] An engineering handbook by ELECTRO-CRAFT CORP(Expanded Third Edition), "DC motors speed controls servo systems"
[7] 조영준, 정일용, 심현승, "DC 소형 모터의 설계 및 해석 기술 개발", 제7회 첨단생산시스템 Workshop 발표 논문집, 1999. 9.
[8] 최영, 정태경, 조성욱, "중소형 모터의 설계/해석 및 평가 모듈 개발", 제7회 첨단생산시스템 Workshop 발표 논문집, 1999. 9.
[9] 임종수 저, "Matlab 완벽가이드 II(응용편)"
[10] Matlab manual, "Getting started with MATLAB (Ver5.3)"
[11] Matlab manual, "Building GUIs with matlab"
[12] Matlab manual, "SIMULINK / Dynamic system simulation for MATLAB"

허 경 무

1979년 서울대학교 전자공학과 졸업. 한국과학기술원 전기 및 전자공학과 석사(1981). 동대학 박사(1989). 현재 단국대학교 공학대학 전자공학과 부교수. 주요 연구분야는 로봇트 제어, 3차원 디스플레이, 시스템 제어, 학습 제어, 디지탈 제어기 설계 및 Simulation, Servo Motor 제어.

조 영 준

1979년 한양대학교 기계공학과 졸업. 한국과학기술원 기계공학과 석사(1981). 동대학 박사(1989). 현재 한국생산기술연구원 생산시스템개발센터 수석연구원. 주요 연구분야는 프랜트 제어 및 응용, 모델링, 학습 제어, 디지털 제어기 설계 및 Simulation, Servo Motor 제어, 반도체 센서응용.

이 은 오

1996년 단국대학교 전자공학과 졸업. 현재 동대학원 석사과정. 관심 분야 시스템 제어, 학습 제어, Servo motor제어, 원격 로봇 제어.