최적제어

최 기 영
(인하대학교 항공우주공학과)

1. 서 론

자동제어의 목적은 먼저 제어대상이 되는 시스템이 불안정할 경우 이를 안정시키고 다음에 이것의 성능을 향상시키는 것이다. 여기서 안정화시킨다는 것은 개념적으로 시스템에 초기조건이 주어졌을 때나 또는 유한한 크기의 입력이 주어졌을 때 시스템의 출력도 유한한 크기로 제한되도록 만드는 것으로 이해할 수 있다. 즉 시스템의 출력이 발산하지 않고 어떤 유한한 범위에 있게 만든다는 것이다. 성능에 대한 기준은 이보다 다양하다. 예를 들어 A점에서 B점으로 이동하는 경우를 보자. 이때 두 점 사이를 최단시간에 이동할 수 있는 시스템이 최적의 성능을 가졌다고 규정해보자. 서울에서 부산사이를 가장 빠른 시간에 갈 수 있는 방법은 로켓을 이용하는 것이다. 하지만 이 경우에는 어느 누구도 이것이 최적의 성능을 가지는 시스템이라고 이야기하지 않는다. 비용면에서 너무나 비현실적이기 때문이다. 그러면 시간은 좀 더 걸리더라도 비용이 적게 드는 방법을 생각해보자. 기차를 이용하면 시간은 앞의 경우보다 더 걸리지만 비용이 훨씬 저렴하다. 따라서 많은 사람이 이것이 최적의 방법이라고 생각할 것이다. 하지만 또 어떤 사람은 4시간 남짓 걸리는 시간이 너무 길어서 비용을 2배정도 더 지불하고 시간을 반 정도로 단축시키고 싶어 할 것이다. 이러한 사람들은 비행기를 이용하는 것이 최적의 방법이라고 생각할 것이다. 이처럼 최상의 성능이라고 하는 것은 관점을 어디에 두느냐에 따라 달라지는 것이고 일반적으로 모든 기준에서 절대적으로 우월한 제어기(해법)는 없다고 말할 수 있다. 최적화라고 하는 것은 보통 몇 개의 인자들을 적당히 섞어서 하나의 성능지수를 만들고 이 성능지수를 최소화(또는 경우에 따라 최대화)시키는 것을 말한다. 여기서는 이러한 최적화 문제에 대한 일반적인 소개와 함께 이러한 방법이 항공기의 제어에 이용된 예를 소개한다.

2. 본 론

2.1. 정적 및 동적 최적화 문제

앞에서 최적화를 이야기할 때는 '무엇(성능지수)을 최적화 시킬 것인가'를 정의하는 것이 중요하다는 것을 보았다. 최적화의 개념은 정적 함수의 최소화 또는 최대화와 맞물려 오래 전부터 알려져 왔다. 인류의 역사와 함께하는 경제문제에서 적은 비용으로 최고의 이익을 실현하는 문제가 가장 전형적인 최적화 문제라고 볼 수 있다. 공학적인 측면에서 볼 때 Newton은 17세기에 이미 최소의 공기저항을 가지는 물체의 형상을 수학적으로 계산하려는 시도를 하였다.[1] 재미있는 것은 Newton이 이 결과를 적용하려고 하는 대상은 선박이나 저속의 비행체이고 여기에 작용하는 공기저항을 공기입자의 운동량의 변화로 해석하려고 했는데 이러한 방법은 배나 비행기의 경우에는 잘 맞지 않는 이론이다. 반면에 이 식은 현대에 와서 실용화된 우주선이나 대륙간 탄도탄 등 극초음속 비행체에 아주 잘 적용이 된다. Newton은 이 문제를 변분법을 사용하여 해석하였는데 이 변분법은 오늘날까지도 최적제어이론의 가장 기본이 되는 방법 중의 하나로 자리하고 있다. 이 외에도 같은 둘레를 가지고 만들 수 있는 도형 중에서 최대 면적이 되는 문제, 일정한 속도로 흐르는 강물에서 건너편에 가장 빨리 도달하기 위해 나아가야 하는 방향 등 많은 문제들이 최적화 문제에 속한다. 이와 함께 주어진 데이터를 분석하기 위해 널리 사용되는 curve fitting 등도 이러한 최적화 문제에 속하는데 이는 모두 정적인 최적화 문제라고 볼 수 있다. 즉 대상이 되는 시스템의 특성이나 상태가 변하지 않는 것이다. 반면에 동적인 시스템에 대한 최적화 문제의 경우에는 시스템의 운동방정식 등 시간에 따라 변하는 요소가 존재한다. 이 분야에서 가장 고전적인 문제 중의 하나는 1696에 John Bernoulli에 의해 제안된 Brachistochrone(그리스어로 '최단시간'의 뜻)문제인데, 이것은 줄에 꿰어진 구슬이 두 점을 잇는 줄을 따라 흘러내릴 때 가장 빨리 목표점으로 가려고 할 때 줄의 모양이 어떻게 되어야 하느냐는 문제이다. 이러한 문제도 앞에서 언급한 변분법을 사용해서

풀 수 있는데 좀 더 복잡하고 실용적인 문제로는 달이나 화성으로 여행할 때 최소의 연료로 가는 방법 또는 최단시간에 가는 방법 같은 문제들이 있다. 이러한 문제들은 일반적으로 다음과 같은 형태의 성능지수를 최소화시키는 것으로 문제를 정형화시킬 수 있다.

$$J = \phi[x(t_f), t_f] + \int_{t_0}^{t_f} L[x(t), u(t), t]\, dt \tag{1}$$

여기서 $u(t)$는 입력, $x(t)$는 시스템의 상태를 나타내는 변수이다. 즉 시스템의 성능을 입력과 상태를 조합한 식을 적분하고 또 그 결과를 최종상태 및 소요시간과 결합한 형태로 표현하였다. 이러한 동적최적화 문제는 간단한 경우가 아니면 일반적으로 해석적인 해를 구하기 어렵다. 따라서 최적화문제가 오래된 문제임에도 비교적 근래에 각광을 받게 된 것은 디지털 컴퓨터의 개발 보급과 관련시켜볼 때 자연스러운 것이다. 이러한 계산기능을 이용해서 복잡한 최적화 문제를 해결하기 시작한 것 중에서 가장 유명한 문제 중의 하나는 최단시간에 일정고도와 속도에 도달하기 위한 비행 조건을 계산하는 것이었다. 이 것은 Bryson과 Denham이 1961년에 F-4기에 대해 수치적인 해를 구하고 그 결과를 1962년 비행시험을 통해서 검증했는데 그림 1에 그 결과를 나타내었다.[2] 단순히 생각하면 가장 빨리 높은 고도로 상승하기 위해서는 수직으로 상승하는 것이 가장 유리할 것 같다. 하지만 제한된 엔진추력으로 고도뿐만 아니라 속도까지도 증가시켜야 하므로 이 것이 최적의 조건은 될 수가 없다. Bryson과 Denham은 이를 계산하고 비행시험을 통해서 검증한 것이다. 계산에서는 F-4 A/B기가 고도 20 km, 마하수 1에 도달하기까지 그림에서 주어진 것 같은 조건을 따라가면 332초에 원하는 목표에 도달할 수 있는 것으로 나타났는데 실제 비행시간에서는 338초가 소요되었다. 이 문제는 두 가지 면에서 놀라운 결과를 제공하는데 우선 그 결과가 기존에 조종사들이 일반적으로 수행하는 방법과는 많이 달랐고 또 이러한 과정을 통해서 얻는 결과가 기존의 방법보다 월등히 우수했다는 것이다. 두 번째로는 그 결과의 정확성이다. 이 문제가 여러 가지 비선형성을 포함하는 복잡한 문제이고 본격적으로 디지털 컴퓨터가 사용되기 이전이었음에도 제한된 계산기능을 가지고 불과 2% 정도의 오차로 결과를 예측했다는 면에서 놀라운 것이다. 이를 두고 Bryson은 '운이 좋았다'고 농담했다.

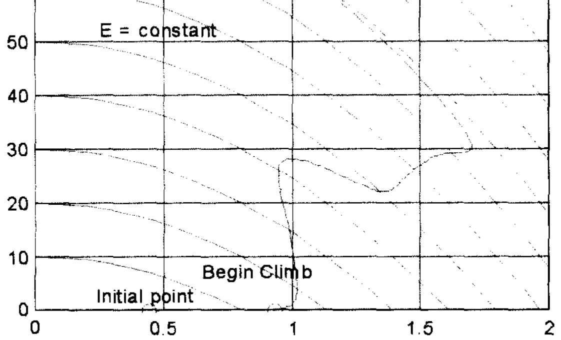

그림 1. 최소시간에 고도 20km, 마하수 1에 도달하기 위한 비행상태 (F-4 A/B)

2.2. 디지털 컴퓨터의 개발과 현대제어

최적제어 이론의 실용화가 디지털 컴퓨터의 개발과 맞물려 있다는 것은 앞에서 언급하였다. 앞의 비행기의 운동을 최적화시키는 문제는 일반적으로 모델을 아주 단순화시키지 않으면 해석적으로 풀기가 불가능하고 따라서 최적제어 문제의 해결에는 필연적으로 (초기의 개발자들에게는) 큰 계산 용량을 가진 컴퓨터가 필요하게 된다. 한편 이러한 큰 계산 용량을 가진 컴퓨터가 개발되면서 일반적인 제어기 설계과정에서 새로운 방법이 도입되었다. 현재에 일반적으로 고전적제어라고 부르는 방법에서는 모델을 입력과 출력 사이의 전달함수로 표현하고 이를 이용해서 시스템을 해석하고 여기에 맞는 제어기를 설계한다. 이 전달함수는 입력과 출력의 Laplace 변환의 비를 나타낸 것으로 원래의 시스템의 운동방정식을 Laplace 변환을 통해 대수 유리식으로 바꿈으로서 해석을 용이하게 할 수 있다는 장점 때문에 널리 사용되었다. 컴퓨터의 개발은 이러한 기존의 방법에서 탈피하여 시스템의 특성을 나타내는 미분방정식(운동방정식)을 수치적분을 사용해서 그대로 풀 수 있게 만들었다. 따라서 시스템을 전달함수 형태로 표현하는 것이 아니고 시간에 대한 1차 연립 미분방정식으로 표현하고 이를 이용하여 제어기를 설계하는데 이것이 현대제어 이론의 기본이 되는 것이다. 이 때 시스템의 (선형)모델은 다음과 같은 벡터 미분방정식으로 주어진다.

$$\begin{aligned} \frac{d}{dt}\mathbf{x} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \end{aligned} \tag{2}$$

여기서 $\mathbf{u}(t)$는 입력 $\mathbf{x}(t)$는 시스템 상태, 그리고 $\mathbf{y}(t)$는 출력을 나타낸다. 한 가지 반드시 언급해야 할 것은 고전적제어와 현대적제어의 차이가 방법상의 우월성을 내포하지는 않는다는 것이다. 소위 말하는 고전적방법으로 제어기를 설계하는 것이 경우에 따라서는 훨씬 직관적이며 효과적일 때가 많다.

2.3. 최적제어의 기본 - LQR

앞에서 몇가지 최적화 문제들에 대해 소개하였는데 이러한 문제들의 경우에는 '제어'라는 개념이 들어가 있지 않다. 좀 더 구체적으로 이야기하면, 우리가 '자동제어'라고 이야기 할 때는 되먹임(feedback) 구조에 대해 이야기 하는 것이 일반적이다. 즉 시스템의 상태를 되먹임 과정에서 이용하여 새로운 입력을 찾아내는 것이다. 되먹임을 사용할 때 제어기를 어떻게 설계하면 주어진 성능지수를 최적으로 만

족하는 시스템이 될까 하는 것이 최적제어의 기본이 되는 것이다. 이러한 최적제어 문제의 해결에서 가장 효시가 되는 것은 1960년에 발표된 Kalman의 논문이다.[3] (혁신적인 결과인데도 멕시코에서 발행되는 논문지에 발표된 것이 흥미롭다.) 위의 식(2)에서 주어진 것 같은 시스템이 초기조건이나 외란의 영향으로 정상상태에서 벗어나 있고 이 것을 다시 정상상태($\mathbf{x}=0$)로 되돌리는데 있어 최적으로 이 작업을 수행하는 문제를 보자. 이 때 최적화의 판단기준이 되는 성능지수는 다음과 같이 정의하자.

$$J=\frac{1}{2}\ \mathbf{x}_f^T\ \mathbf{S}_f\ \mathbf{x}_f+\frac{1}{2}\int_{t_0}^{t_f}(\ \mathbf{x}^T\ \mathbf{Q}\ \mathbf{x}+\ \mathbf{u}^T\ \mathbf{R}\ \mathbf{x})\,dt \qquad (3)$$

즉 시스템의 반응속도가 느리면 $\mathbf{x}^T\mathbf{Q}\mathbf{x}$항에 의해 성능이 감소하고, 또 제어입력을 너무 많이 사용하면 $\mathbf{u}^T\mathbf{R}\mathbf{x}$항에 의해 성능이 감소하게 되므로 너무 과도한 제어 입력을 사용하지 않으면서 $\mathbf{x}$를 가급적 빨리 줄여 나가도록 설정되었다. 여기서 $\mathbf{Q}$와 $\mathbf{R}$은 상대적인 중요도를 나타내는 가중 행렬이다. Kalman은 선형시스템의 경우에 이러한 성능지수를 사용하면 제어기 전체가 선형으로 구성될 수 있다는 것을 보였다. 즉 빠른 시간에 제어입력을 재 계산하는 것이 가능하므로 실용화가 가능한 제어기를 구성할 수 있는 토대가 마련된 것이다. 여기에서 좀 더 나아가서 t_f가 아주 큰 경우에는 제어기가 시불변이 된다는 것을 볼 수 있다. 즉 위의 성능지수를 최소화시키는 부최적(suboptimal) 제어기가 다음과 같은 간단한 형태로 구성된다.

$$\mathbf{u(t)}=-\ \mathbf{Kx(t)} \qquad (4)$$

이러한 제어방식을 Linear Quadratic Regulator(LQR)이라고 한다.

2.4. 추정기와 최적제어

위의 LQR의 식을 다시 보면 제어입력이 시스템의 상태에 의존한다는 것을 알 수 있다. 시스템의 상태를 나타내는 상태변수벡터 $\mathbf{x}$의 크기는 미분(운동)방정식의 차수와 일치하는데 보통 센서로 측정하는 출력의 개수보다 많다. LQR은 출력을 되먹임에 사용하는 것이 아니라 상태변수벡터를 사용하기 때문에 실용적인 문제에 있어서 제한될 수밖에 없다. 이러한 문제를 해결하기 위해 추정기가 개발되었다. 이 추정기는 제한된 출력을 이용하여 필요한 상태변수벡터를 구성하는 것이다. 예를 들어 항공기에서 측정하는 것이 위치뿐이라고 하고 이 위치센서는 잡음도 없고 아주 정확한 신호를 낸다고 하면 상태변수에 속하는 속도는 이 위치를 미분해서 구할 수 있다. 이때 이 미분기가 추정기의 역할을 담당하는 것이다. 실제로는 센서가 잡음이나 상수오차 등으로 인해 부정확한 값을 내게 되므로 단순한 미분기를 사용해서 상태를 추정할 수는 없게 된다. 하나의 해결 방법은 위의 LQR과 유사한 성능지수를 정의하고 측정된 출력과 추정된 상태에서 재구성한 출력의 차이를 최소화시키면서 상태를 추정하게 된다. 이러한 추정기와 위의 LQR을 사용하면, 그림 2에서처럼 센서에서 측정된 출력을 추정기의 입력으로 사용하고, 이 추정기에서 추정된 상태변수 벡터에 LQR 이득을 곱해서 제어 입력을 만들어서 다시 시스템에 입력함으로써 최적의 성능을 가지는 실용적인 제어기가 만들어 질 수 있는 것처럼 보인다. 1970년대에는 이러한 제어기가 아주 실용적이며 광범위하게 사용될 수 있을 것처럼 보였다. 하지만 이러한 시스템이 가지고 있는 문제점이 몇 가지가 있었는데 첫 번째는 고전적제어기보다 더 복잡하다는 것이고 두 번째는 강건성이 보장이 되지 않는다는 것이다.

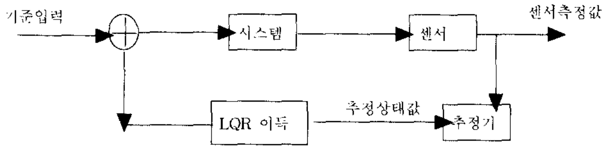

그림 2. 추정기와 결합된 LQR

2.5. 강건제어

제어기를 개발하는 과정에서 가장 기본 단계에 해당하는 것은 제어 대상이 되는 시스템의 수학적 모델을 구하는 것이다. 비행기나 로켓 등의 복잡한 시스템을 쉽게 다룰 수 있는 범위에서 표현하기 위해서는 필연적으로 가정들을 부가하게 된다. 이러한 가정들을 통해서 모델의 차수를 줄이고 선형화시킬 수 있는 것이다. 따라서 이 과정에서 수학적으로 표현된 모델과 실제의 물리적인 시스템의 특성사이에는 차이가 존재하게 되는데, 가정들을 사용해서 단순화시킨 모델에 대해 설계한 제어기를 실제 시스템에 적용하였을 때는 그 효용성이 설계에서 제시한 것보다 떨어질 수밖에 없다. 심한 경우에는 안정성마저도 보장이 되지 않아 불안정해 질 수 있는 것이다. 이처럼 모델에 불확실성이 존재할 때 전체 제어 시스템의 안정성과 성능이 보장되도록 제어기를 구성하는 것을 강건제어(robust control)라고 한다. 앞에서 언급하였던 추정기와 결합된 LQR은 이러한 강건성이 보장되지 않아서 성능이 제대로 발휘될 수 없는 경우가 자주 발견되었고 그 결과 1980년대부터는 이러한 강건성을 보장하는 제어기를 설계하는 것이 하나의 큰 축으로 자리 잡았다.

3. 결 론

이 글에서는 최적화의 개념에서 출발하여 정적 및 동적 최적화 문제를 소개하고 이를 발전시켜 최적제어 문제까지 언급하였다. 아울러 1960년대 및 70년에 활발히 전개되었던 최적제어 이론이 1980년대에 오면서 강건제어 이론과 결합되어 나타나는 과정까지를 설명하였다. 앞에서 언급하였듯이 최적화시킨다는 것은 어떤 성능지수를 정의하고 그 것을 최대화 또는 최소화시키는 문제로 정식화된다. 여기에서

는 성능지수를 주로 자승형태(quadratic form)로 주어지는 문제들에 대해 다루었는데 이 외에도 많은 형태의 최적화 문제들이 존재한다. 예를 들어 강건제어의 한 부분인 H_∞제어기의 경우에는 시스템의 특성의 일부를 나타내는 H_∞ norm을 성능지수로 설정하고 이것을 최소화시키는 제어기를 찾는 것을 목적으로 하면 이 과정에서 주어진 범위에서 강건성을 보장하는 제어기가 개발되는 것이다.

최적제어기를 구성하는 과정에서는 많은 계산이 필요하다. 따라서 좋은 계산방법을 개발하는 것이 또한 필수적인데 여기서는 이러한 계산방법에 대해 지면관계상 다루지 못하였다. 현재에는 많은 제어공학자들은 기본계산을 위해서는 별도의 프로그램을 쓰지 않고 MATLAB[4]을 사용하고 또 MATLAB의 프로그래밍 기능을 사용하여 복잡한 계산까지도 수행한다. MATLAB에서 제공하는 Optimization Toolbox나 Control Toolbox, 또 Robust Control Toolbox를 사용하면 많은 경우에 별도의 프로그래밍없이 빠른 시간에 계산을 수행할 수 있으므로 제어기를 설계하는 엔지니어들이 계산과정보다는 물리적인 현상을 이해하고 분석하는데 집중할 수 있다.

이러한 최적제어이론을 사용해서 개발된 제어기는 일반적으로 고전적제어기보다 좀 더 복잡한데 컴퓨터 및 전자부품의 비약적인 발전으로 이러한 것에 대한 제약이 줄어들면서 그 활용도가 점점 넓어지고 있다. 항공기나 미사일의 제어에 이러한 최적제어 이론을 적용시킨 다양한 결과들을 기본 이론과 함께 참고문헌 [5]에서 확인할 수 있다.

참 고 문 헌

[1] A. E. Bryson, Jr., Y.C. Ho, *Applied Optimal Control*, Halsted Press, 1975

[2] A. E. Bryson, Jr., *Dynamic Optimization*, Addison Wesley, 1999

[3] R. E. Kalman, "Contributions to the Theory of Optimal Control," *Bol. de Soc. Math. Mexican*, 1960

[4] http://www.mathworks.com

[5] C.-F. Lin, *Advanced control Systems Design*, Prentice Hall, 1994