비계수 모델에 대한 1차 안정화기 전체 셋의 수치 해

임연수*, 김려화*, 김근식**, 김영철*
*충북대학교, **아주자동차대학교

Numerical Solution of All Stabilizing First Order Controllers for Non-Parametric Model

Yearnsu Lim*, Lihua Jin*, Keunsik Kim**, Young Chol Kim*
*Chungbuk National University, **Ajou Motor College

Abstract - This paper presents a numerical solution of first order controller design. The complete set of first order controllers that stabilizes for a LTI plant solely determined from the knowledge of the frequency response and the number of unstable poles of the plant. An illustrative example is given to describe the design procedure and exhibit graphical displays of the feasible regions of the design parameters.

1. 서 론

산업현장은 고정된 구조를 갖는 저차 제어기(예: PID제어기 또는 1차 안정화기)의 사용을 항상 필요로 한다. 그것들은 구조가 간단하면서도 효과적인 제어를 가능하게 하기 때문이다. 그러한 제어기는 주파수 영역 데이터나 입수 가능한 수학적인 모델(전달함수 또는 상태 모델)들을 이용하여 설계를 할 수 있다. 1차 안정화기의 경우, 주어진 모델의 전달함수에 대해 폐 루프 시스템을 안정하게 하는 제어기의 전체 셋을 구하는 방법들이 이미 개발되었다[1-3]. 그러나 해석적인 모델을 구하는 것은 실용적인 시스템 식별 방법을 사용하지 않고서는 구하기가 어렵다.

최근에 플랜의 주파수 응답과 RHP에 놓인 극점의 개수만으로 안정도와 다양한 성능(게인여유, 위상여유, H_∞여유)을 만족하는 1차 안정화기의 셋을 구하는 새로운 방법이 소개되었다[4]. 그러나, 그 방법을 소프트웨어적으로 구현하는 과정에서는 많은 어려움에 봉착하게 된다. 본 논문에서는 그 구현을 가능케 하는 수많은 수치 해 중 하나를 제시하고자 한다.

2. 비 계수 모델에 대한 1차 안정화기의 전체 셋 계산

LTI 플랜트와 1차 안정화기를 포함하는 피드백 제어 시스템은 그림 1과 같다.

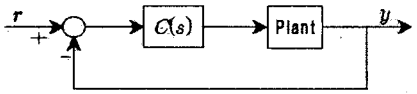


그림 1. 단위 피드백 시스템

제어기의 계수는 다음과 같이 정한다.

$$C(s)=\frac{x_1 s+x_2}{s+x_3}. \tag{1}$$

설계를 위해 알고 있는 것들은 다음과 같다.

$$\begin{cases} P(j\omega)=P_r(j\omega)+jP_i(j\omega), \quad \text{for}\ \omega\in[0,\infty] \\ \text{RHP에 놓인 극점의 개수 } p_r \end{cases} \tag{2}$$

주어진 데이터를 이용하여 다음에 해당하는 2개의 직선과 1개의 곡선을 구할 수 있다.

$$x_3+x_2P(0)=0 \tag{3}$$

$$1+x_1P(\infty)=0 \tag{4}$$

$$\begin{cases} x_1(\omega)=\dfrac{1}{|P(j\omega)|^2}\left(\dfrac{P_i(\omega)}{\omega}x_3-P_r(\omega)\right) \\ x_2(\omega)=-\dfrac{1}{|P(j\omega)|^2}\left(P_r(\omega)x_3+\omega P_i(\omega)\right) \end{cases} \tag{5}$$

위의 식들 중 식 (4)의 경우 플랜트의 수학적인 모델의 최고차항 계수가 같을 때($d_n=n_n$)에만 적용해야 한다. 이는 주어진 주파수 응답의 크기 선도에서 고주파영역의 기울기가 0에 가까운지 살펴보면 판단할 수 있다.

제어기의 전체 셋을 구하는 과정의 순서도를 그림 2에 나타내었다. 2번째 과정인 2개의 직선과 1개의 곡선을 구하는 것은 [4]의 내용을 참조하면 쉽게 할 수 있다. 하지만 그 이후의 과정들을 구하는 것은 쉽지 않다. 계산 과정에서 x_1, x_2의 이산적인 좌표 데이터로부터 각 선들의 교차점, 최소단위 폐 경로, 폐 경로 내의 점을 구하게 된다. 이러한 것들은 사람이 그림으로 도시한 후 판단한다면 쉬운 일이지만, 컴퓨터가 수치 데이터만 주어진 상태에서 판단하기는 어렵기 때문이다.

각 과정에 대한 설명은 다음의 예제를 통하여 하겠다.

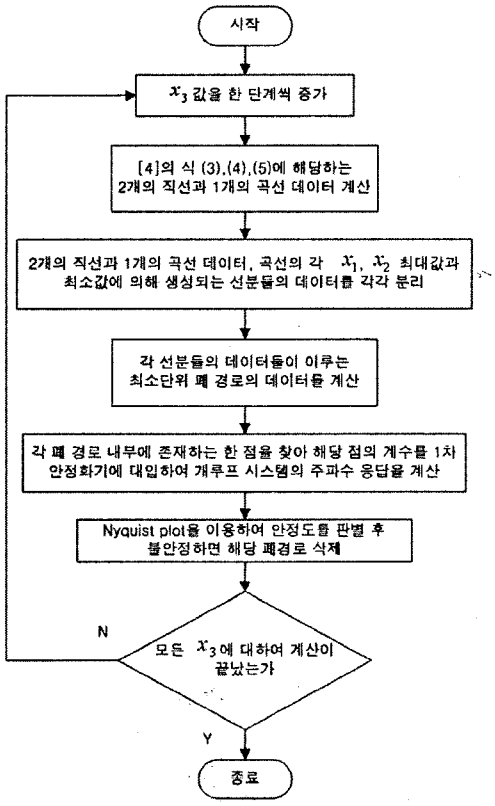


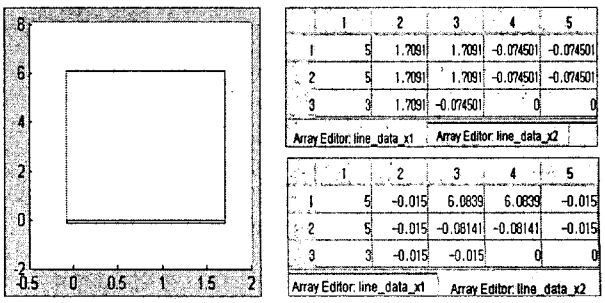
그림 2. 수치 해의 순서도

예제 1 [1]과 [4]에서 다루는 예제를 사용하면, 시스템은 $p_r=0$ 이고 주어진 주파수 응답은 다음과 같이 가정한다.

$$P(j\omega)=\{P(j\omega):\omega\in(0,10)\ \text{sampled every } 0.01\}$$

우선 x_3의 값을 0.2로 고정한 후 아래의 과정들을 통해 안정 영역을 구한다.

1. 선분들의 데이터 분리 : 우선 곡선 데이터의 최소값과 최대값을 이용하여 계산 영역의 범위를 제한한다. 계산 영역이 더 넓은 경우, 곡선의 끝부분에서 그것에 의해 생성되는 닫힌 경로가 완성이 되지 않을 수 있기 때문이다. 이번 예제에서는 $d_n\neq n_n$이므로 1개의 직선만 사용해야 한다. 영역의 경계선을 직선으로 분할하면 그림 3의 결과가 나온다.



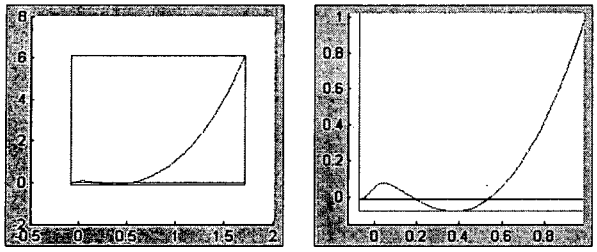
	1	2	3	4	5
1	5	1.7091	1.7091	-0.074501	-0.074501
2	5	1.7091	1.7091	-0.074501	-0.074501
3	3	1.7091	-0.074501	0	0

	1	2	3	4	5
1	5	-0.015	6.0839	6.0839	-0.015
2	5	-0.015	-0.08141	-0.08141	-0.015
3	3	-0.015	-0.015	0	0

그림 3. 계산 영역의 경계와 직선에 의해 생성된 선분 데이터

마지막으로 곡선을 이용하여 각 선분들을 분할하면 그림 4의 결과가 나

온다.



Numeric format: short0 Size: 16 by 101

	1	2	3	4	5	6	7	8
1	3	-0.074501	-0.074501	0	0	0	0	
2	3	-0.074519	-0.074501	0	0	0	0	
3	3	0.19469	-0.074501	0	0	0	0	
4	47	-0.074501	-0.073967	-0.07305	-0.071767	-0.070123	-0.068123	-0.065
5	4	0.36083	-0.074501	-0.074501	0	0	0	
6	11	0.19469	0.20837	0.22497	0.24231	0.26038	0.27919	0.298
7	4	1.7091	1.7091	0.3729	0	0	0	

Array Editor: line_data_x1 | Array Editor: line_data_x2

Numeric format: short0 Size: 16 by 101

	1	2	3	4	5	6	7	8
1	3	-0.014975	-0.015	0	0	0	0	
2	3	-0.014977	-0.014975	0	0	0	0	
3	3	-0.015	-0.015	0	0	0	0	
4	47	-0.014975	-0.014905	-0.014774	-0.01457	-0.014269	-0.013839	-0.013
5	4	-0.08141	-0.08141	-0.015	0	0	0	
6	11	-0.015	-0.024175	-0.034594	-0.04457	-0.053671	-0.062254	-0.069
7	4	-0.015	-0.08141	-0.08141	0	0	0	

Array Editor: line_data_x1 | Array Editor: line_data_x2

그림 4. 각 직선과 곡선에 의해 분할된 선분들의 데이터

2. Root Invariant Region 계산 : 각 선분들의 데이터 양 끝이 만나는 점들을 따로 저장한 후, 한 점으로부터 다른 점으로 이동 가능한 모든 경로의 데이터를 생성한다. 이 데이터는 모두 폐 경로를 나타낸다. 생성된 폐 경로 데이터 중 중복이 되는 것들은 1개의 데이터만 남기고 나머지는 모두 삭제한다. 남은 폐 경로들 중 해당 경로에 없는 선분 위의 점이 경로 내부에 존재하는 경우 해당 경로를 삭제하면, 연산 결과 그림 5와 같이 다른 폐 경로를 포함하지 않는 최소단위 폐 경로만 남게 된다. 이 최소단위 폐 경로의 내부 영역이 Root Invariant Region이 된다.

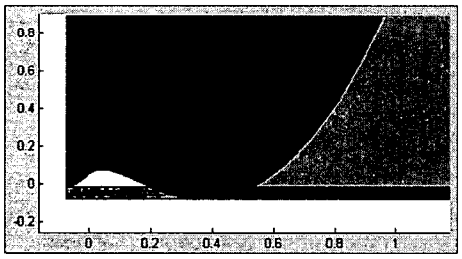


그림 5. 계산된 Root Invariant Regions

3. Root Invariant Region 내부의 점에 해당하는 제어기의 계수에 맞게 개 루프 시스템의 주파수 응답 계산 : 각 영역을 나타내는 폐 경로의 선분이 만나는 점을 기준으로 각 선분 위의 가장 근접한 점 2개를 선정한 후, 그 점들을 통해 영역 내부의 점을 구한다. (x_1, x_2, x_3)에 맞는 제어기의 주파수 응답 $C(j\omega)$를 구하고 이를 통해 개 루프 시스템의 주파수 응답을 계산한다.

4. Nyquist 선도를 이용한 안정도 판별 : 개 루프 시스템의 주파수 응답을 이용한 Nyquist선도가 실수축에서 -1보다 작은 구간과 교차하는 부분을 해석하여 안정도를 판별한다. 불안정한 폐 경로를 모두 삭제하면 그림 6에 해당하는 부분만 남는다. 이 영역이 안정 영역이 된다.

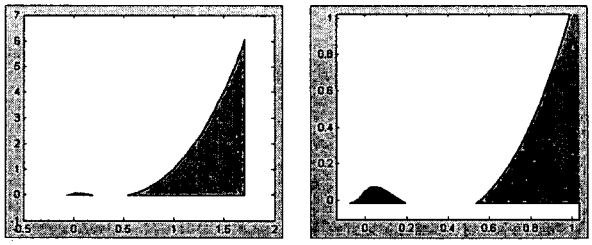


그림 6. $x_3 = 0.2$ 일 때의 안정 영역

앞의 방법을 이용하여 각 x_3에 대한 Root Invariant Region을 구하면 해당 x_3구간에 대한 1차 안정화기의 전체 셋을 구할 수 있다. x_3를 0.01에서 0.54까지 증가시키면서 구한 셋을 그림 7에 도시하였다.

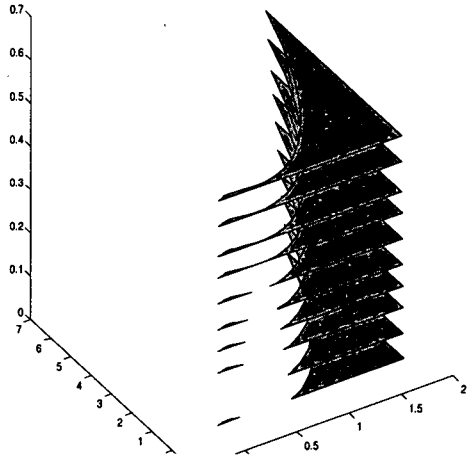


그림 7. $0.01 \le x_3 \le 0.54$ 에서의 1차 안정화기 전체 셋

3. 결론

본 논문에서는 비계수 모델에 대해 안정도를 만족하는 1차 안정화기의 전체 셋을 구하는 수치 해를 제시하였다. 앞서 [4]에서 소개된 방법은 복잡한 모델링의 과정이 필요 없으므로 매우 유용하다. 하지만 주어진 수치 데이터만으로는 풀기 어려운 그래픽적인 해석이 필요하기 때문에 그것에 사용할 수 있는 여러 방법 중 하나를 여기에서 소개하였다. 한 가지 아쉬운 점은 주파수 응답 데이터가 유한하기 때문에 nyquist 선도가 충분히 그려지지 않아 안정도 판별에 오류가 있을 수 있는 점이다. 만약 데이터가 충분하다면, 이러한 문제점은 개선이 될 것이라 생각된다. 현재는 안정도와 함께 다양한 성능(게인여유, 위상여유, H_∞여유)을 만족하는 1차 안정화기의 전체 셋을 구하는 과정을 구현 중에 있다.

감사의 글

본 연구는 한국과학재단 특정기초연구 (과제번호: R01-2006-000-10811-0) 지원으로 이루어진 연구임.

[참 고 문 헌]

[1] R.N. Tantaris, L.H. Keel, and S.P. Bhattacharyya, "Stabilization of Continuous Time Systems by First Order Controllers," *Proc. of the 10th Mediterranean Conference on Control and Automation*, Lisbon, Portugal, July 9-12, 2002.

[2] R.N. Tantaris, L.H. Keel, and S.P. Bhattacharyya, "Gain/Phase Margin Design with First Order Controllers," *Proc. of the American Control Conf.*, Denver, Colorado, 2003, pp. 3937-3942. 2005, Seville, Spin, Dec. 2005.

[3] R.N. Tantaris, L.H. Keel, and S.P. Bhattacharyya, "H_∞ Design with First-Order Controllers," *Proceedings of the 2003 IEEE Conference on Decision and Control*, Maui, HI, December 9-12, 2003.

[4] L.H. Keel, and S.P. Bhattacharyya, "Direct Synthesis of First Order Controllers from Frequency Response Measurements," *Proc. of the American Control Conf.*, Portland, OR, USA, June 8-10, 2005.