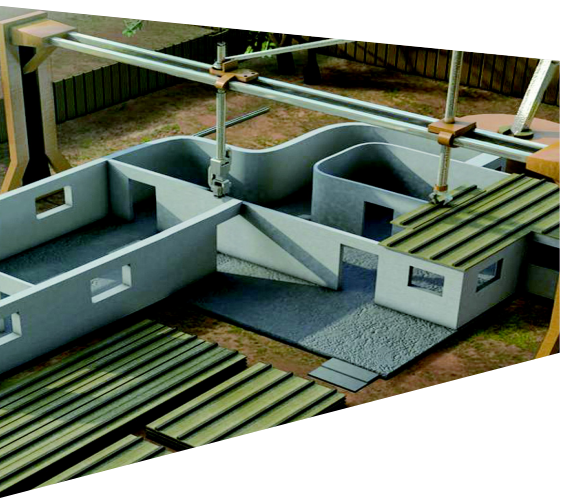


모델축약: 프로그램다이어트

Model Reduction Techniques

이 용 제

㈜창민우구조컨설탄트 소장

1. 개요

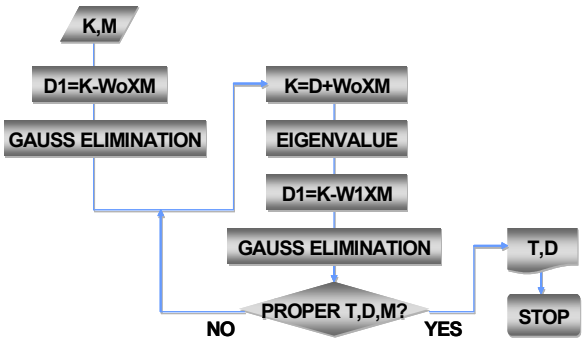
구조해석 프로그램은 해석 대상의 다양화, 해석 결과의 정교함, 그리고 보다 이해하기 쉬운, 때론 과도해 보이기까지 한 시각화 등 많은 발전을 거듭해 오고 있다. 그 역할에 있어서도 과거 수계산으로 일 단락한 구조해석 결과를 2D 기능의 해석 프로그램으로 검토하는 정도였으나 이제는 구조해석하면 프로그램상의 모델링을 떠올릴 정도로 다양한 구조물을 대상으로 특화된 다양한 프로그램들이 널리 쓰이고 있다. 구조해석 프로그램들의 이러한 대담한 약진은 기본적으로 과거와는 차원이 다른 고성능 컴퓨터의 발전에 힘입은 바가 크다. 즉, 프로그램 자체의 무게(혹은 복잡함)를 보다 빠른 처리속도를 지원하는 하드웨어에 의해 상쇄하는 개념인 것이다. 그러나 여전히 구조공학의 여러 분야에서 속도의 문제는 비용의 문제를 야기시키며 풀리지 않는 숙제로 남아있다. 예를 들어 수천번 이상의 시뮬레이션을 요하는 확률론적 접근의 연구 등에서 빠른 해석은 전체 프로젝트의 성패를 좌우할 수 있는 열쇠가 될 수 있다. 실무의 경우, 우리사가 설계한 123층 규모의 롯데월드타워의 경우가 좋은 예가 될 것이다. 국지적인 구조변경에도 전체 모델을 처음부터 다시 해석해야 했고 워낙 자유도가 높은 구조물이다 보니 그 해석 시간 또한 만만치 않게 소요되었다. 특히 최근 이 건물의 구조건전도평가시스템(Structural Health Monitoring System) 용역을 수행하면서, 전산모델의 무게를 실감하게 된다. 즉, 인공신공망(ANN: Artificial Neural Network)을 이용한 손상탐지(Damage Detection) 기술의 적용을 시도함에 있어, “런타임”의 장벽에 막혀 모델블럭을 무한정 세분화하기가 곤란한 상황에 처하기도 한다. 마치 그 옛날 도스(DOS)시절 프로그램을 돌리면서 만났던 상황의 데자뷰이며, 하드웨어의 눈부신 발전도 프로그램의 복잡성의 증가와 평행선을 달려온 형국이다. 이런 문제를 해결하기 위해 많은 연구자들이 프로그램을 가볍게 다이어트하기 위한 다양한 전략들을 구상해 왔다. 소위 모델축약(Model Reduction)이라 불리는 다이어트 트레이너들의 자취를 따라가 보았다.

2. 다이어트 전략들

지금으로부터 약 50년전, R. Guyan(1965)에 의해 미국항공우주학회지에 매우 획기적인 아이디어를 담은 논문 한편이 실린다. {F}=[K]{x}의 공식에서 강성 매트릭스를 블록화 하여 “불필요한” 부분을 제거(Truncation)해 버린 것이다. 이 매직은 불과 대여섯개의 수식으로 완벽하게 증명되고 있다. 게다가 모델축약이라는 본문의 의도를

충실히 재현하듯 논문 전체의 길이도 반페이지에 불과하다. 사실 이런 중요한 연구가(구조공학 혹은 토목학회지가 아닌) 항공우주학회지에 실린 것은 우연이 아닌 듯 하다. 이 분야는 구조제어(Structural Control)라는 다소 복잡한 수단을 필수적으로 탑재해야 했기 때문이다.

이 기술을 이해하기 쉬운 예제와 더불어 구조공학에 소개하며 발전시킨 사람은 "Structural Dynamics" 교과서의 저자로 잘 알려진 M. Paz(1983)이다. 동역학 시스템에의 적용(아래 그림)과 더불어 이 이론은 그가 저술한 교과서의 한 챕터를 차지하고 있다.



이후 많은 연구자들이 다양한 아이디어를 쏟아내고 있다. 그 추세를 보자면 먼 과거의 수학이론을 끄집어 내어 응용하는 경향이 강하며, 방법에 있어서도 모델 일부의 제거라기보다는 축약(Condensation)에 가깝다.

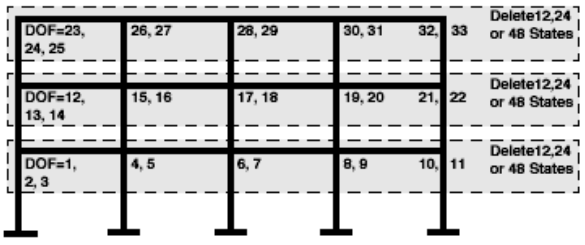
사실 완벽해 보이는 Guyan의 유도식은 치명적인 단점을 내포하고 있다. 즉, 도대체 시스템의 자유도 중 무엇이 불필요한 부분인가를 판단하는 문제이다. 게다가 높은 자유도의 시스템에서는 마음의 결정을 하였어도 그 복잡한 강성도 매트릭스에서 제거할 부분을 잡아내는 일 또한 만만치 않은 작업이 될 것이다. 따라서 제거할 부분을 신속하고 효율적으로 결정하여 모델 축약의 목적을 달성하는 나름의 원칙들이 제안되기 시작하였다.

건축구조공학에 적용된 방식중 하나는, 시스템에서 고차주파수에 해당하는 부분은 전체 시스템에 미치는 영향이 적을 것이라는 가정에 기반 한다. 이 부분을 골라내는 것은 약간의 선형대수적 조작에 의해 쉽게 성취될 수 있다. 또한 이를 응용, 고차주파수에 해당하는 상태의 미분치가 영에 빠르게 수렴한다는 가정을 이용할 수도 있다.

다른 아이디어는 시스템의 상태공간(State-space)에서 그래미안(Gramian)이라는 비 물리적 인덱스를 이용하는 것이다. 그래미안은 제어성 그래미안(Controllability Gramian)과 관측성 그래미안(Observability Gramian)으로 나뉘는데, 이 둘의 특성을 "균형 있게" 구현한 Balanced Gramian을 이용한다. 즉, 시스템에서 어떤 상태가 더 중요하고 중요한지를 그래미안을 통해 판단한 후 해당 부위를 남기거나 지우는 과정을 거친다.

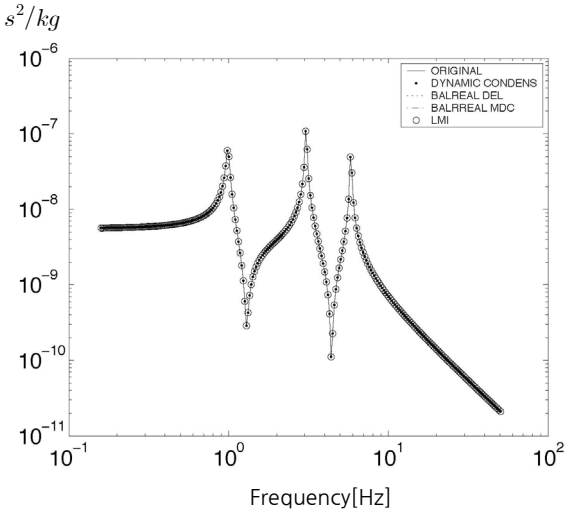
그래미안을 이용하여 시스템의 중요 부위를 남기는 방법은 편리할 뿐만 아니라 축약된 모델의 충실도(Fidelity)가 높은 것이 사실이다. 그러나 여기에도 몇몇 단점들이 존재한다. 이는 위의 방법들이 제거가 아닌 축약의 성격을 띠고 있다는 점에서 문제의 단초를 유추할 수 있겠다. 먼저, 앞에 언급한 선형대수적 조작을 거침으로서 시스템의 인풋과 아웃풋을 연결하는 강성매트릭스 및 질량매트릭스의 배열이 뒤섞이게 된다. 즉, 이 상태에서는 더 이상 구조체상 관심 부분의 물리적 추적이 불가해 진다는 것이다. 결국 이 과정을 통해 어떤 부분이 제거되는지 알기 어려워진다. 둘째는 State-space Realization 과정에서 시스템은 주파수영역(Frequency domain)으로 구현되며, 이는 탄성범위에서의 거동만을 고려할 수 있다는 한계를 뜻한다. 이를 극복하기 위한 이론들이 제시되고는 있지만, 그 복잡성으로 인해 오히려 다이어트라는 본래의 목적을 훼손할 수 있는 것이다.

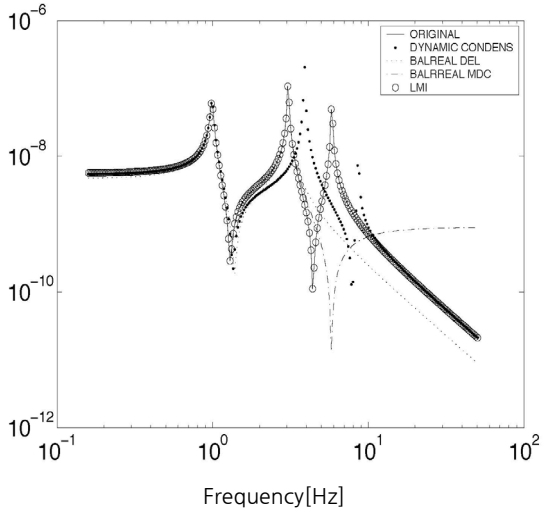
보다 근본적 해결방안으로서 선형행렬부등식(Linear Matrix Inequalities, LMI)을 이용한 연구가 제안되었다(Boyd et al. 1994, Lee and Johnson, 2004). 이 방법이 의도하는 바는 시스템 구성 매트릭스의 블록화 및 블록별 축약을 통해 그 토폴로지(Topology)의 손상을 최소화 하는데 있다. 예를 들어 다음 그림과 같은 3층짜리 간단한 구조의 모델축약을 생각해 보자.



앞서 설명한 바와 같이, 기존의 방식을 사용할 경우 부딪히게 될 문제는 두 가지이다: 1) 모델의 어디를 잘라야 하는가, 2) 시스템의 어디가 잘려나가는가.

LMI를 이용하면 각 층의 상태를 블록화 하는 것이 가능하다. 이후 각각의 블록 내에서 기여도가 작은 부분을 선별하여 축약과정을 진행하면, 모델축약에 관한 엔지니어의 의도를 적극 반





영하면서도 충실도가 높은 결과를 얻을 수 있다. 다음의 그래프는 각 축약 방법별 주파수 증폭 정도를 나타내고 있다.

첫 번째 그래프는 본래 66개의 상태(State) 중 시스템 반응에 영향이 적을 것으로 판단되는 회전 자유도에 해당하는 12개의 상태를 제거한 전달함수의 결과이다. 전 주파수 영역에 걸쳐 본래 시스템의 반응과 거의 차이를 보이지 않는다. 두 번째 그래프는 여기에 추가로 12개의 상태를 제거한 결과이다. LMI 방식을 제외한 모든 모델 축약 기법의 충실도가 떨어진 모습을 볼 수 있으며, 특히 고주파 영역에서 그 정도가 심하였다. 다시 말해, LMI 모델축약 기법을 이용함으로써 시스템의 크기는 현저히 줄이면서도 원래 시스템의 거동과 매우 유사한, 즉 충실도가 높은 모델을 구현할 수 있음을 발견하였다.

3. 결어

요요 없는 다이어트방법들이 고안되어 왔듯이, 전산모델의 축약방식 또한 매우 다양하며 또 그때그때 용도에 따라 적용 방식도 달라진다. 마지막에 소개한 LMI 모델축약 기법 또한 결코 만능의 방법은 아니며, 더 많은 연구를 요하고 있다. 우선 State-space Realization을 거치는 다른 방법들과 마찬가지로 선형모델에 국한되는 방법이라는 점이다. 하지만 보다 도전적인 문제는 이 방법의 이해와 프로그래밍, 그리고 해석시간이 만만치 않다는 점이다. 필자가 학위과정이던 시절, 그리 복잡하지 않은 중층 규모의 구조물을 이 방법으로 축약하기 위해 MATLAB으로 작성한 프로그램이 일주일 넘게 돌아갔던 기억이 있다. 차라리 모델축약 없이 해석했더라면 1시간도 안 걸렸을 모델이었다. 이를 본 지도교수는 “가라앉는 배에서 무엇을 건질 것인지 생각해 보라”는 의미심장한 조언을 해주셨고, 그길로 수년간 진행하던 논문 테마를 부랴부랴 변경했던 가슴 아픈 기억이 있다. 이 부분은 날씬한 프로그램을 열망하며 더 다양하고 새로운 아이디어를 가진 연구자의 손에 의해 가까운 미래에 해결될 것이라 믿는다.

4. 참고문헌

1. Guyan R.J. (1965) Reduction of Stiffness and Mass Matrices, AIAA *Journal*, 3(2), p.380.
2. Paz M. (1983) Practical Reduction of Structural Eigenproblems, Journal of Struct. Eng., ASCE, 109(111), pp.250--2599.
3. Boyd S., Ghaoui L.E., Feron E. and Balakrishnan V. (1994) Linear Matix Inequalities in System and Control Theory, SIAM, Philadelphia, USA.
4. Lee Y. and Johnson E. (2004) LMI Model Reduction for Modeling and Control of Structures, 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Albuquerque, NM, USA.