S1-61
N91-22814

A Fast, Reliable Algorithm for Computing Frequency Responses of State Space Models

Matt Wette
Jet Propulsion Laboratory, Caltech

Abstract

Computation of frequency responses for large order systems described by time-invariant state space systems often provides a bottleneck in control system analysis. In this talk we show that banding the A-matrix in the state space model can effectivly reduce the computation time for such systems while maintaining reliability in the results produced.

Introduction to the Problem

Consider the following realization of some transfer function $G(j\omega)$:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

where $x \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$. The relationship of the realization to the transfer function is given by

$$G(j\omega) = C(j\omega I - A)^{-1}B + D$$

In control system design the computation of the frequency response plays an important role in frequency-based design methods. For medium sized problems the order of $x(t)$ may be in the hundreds while $G(j\omega_k)$ must be computed for hundreds of values of ω_k. It has not been uncommon for a frequency response calculation to require hours of CPU time. Thus, efficient and reliable algorithms for this computation are needed for handling large order systems.

Typical Approach

A typical approach to computing frequency responses for state space systems is to first perform a state transformation on the realization to bring the A-matrix into some reduced form and then solve the appropriate system of linear equations for each frequency point.

Computational Issues

The above algorithm for computing frequency responses involves two issues: efficiency and sensitivity. A potential bottleneck in computing frequency responses is the solution of $(j\omega_k I - \tilde{A})X = B$ for X. Efficient computation is accomplished by reducing A to some form $\tilde{A}$ which allows efficient solution of the above equation. Another issue is that of sensitivity. The transformation process takes place in finite precision arithmetic and hence will change, to some degree, the properties of the transfer function which the realization represents. It is important, therefore, to consider the numerical properties of the transformation.

Sensitivity of the Transformation

The effect of numerical computations in the presense of finite precision arithmetic can be treated in terms of sensitivity of the coefficient matrices. Tranformations which do not increase sensitivity to state transformations are termed *well conditioned.* Ill-conditioned transformations can and usually do significantly increase the sensitivity of the coefficient matrices to small perturbations. Presense of this sensitivity is often an indication of a numerically unstable algorithm.

Efficient Solutions to $(j\omega_k I - A)X = B$

As stated, efficient solution of the above equation is usually accomplished by reducing A to some special form. Consider the case where $m = 1$ (i.e., $B \in \mathbb{R}^{n\times 1}$). Then for A in general form, solution of the equation requires $\mathcal{O}(n^3)$ floating point operations (or flops). For $\tilde{A}$ in Hessenberg form or Schur form, where $\tilde{A}$ is "nearly" upper triangular, solution of the equation requires $\mathcal{O}(n^2)$ flops. The transformation to produce these forms is known to be extremely well-conditioned. For $\tilde{A}$ in diagonal form or Jordan canonical form, solution of the equation requires $\mathcal{O}(n)$ flops. However, the transformation producing these forms may be very ill-conditioned, leading to a reduced form which no longer accurately represents the tranfer function of interest. A good compromise is to seek some compromise between the upper-triangular and diagonal forms. This leads reduced forms in which $\tilde{A}$ is upper-triangular and block diagonal or upper-triangular and banded. We feel the banded form is a better strategy since it is a simpler structure to work with. The banded matrix is characterized by its order and bandwidth; the block diagonal form is characterized by order, block sizes, and block order. Due to the simpler structure of the banded matrix, algorithms based on banded matrices can be adopted to vector hardware architectures in a nicer way.

A New Banding Algorithm

The new banding algorithm uses several steps. First the matrix A is reduced to real Schur, or quasi-upper-triangular, form A_s. Then an ordering algorithm is applied to order the eigenvalues appearing on the diagonal of A_2 in a way that will aid the next step in producing a small bandwidth. The transformations associated with the first two processes is very well conditioned. The third step involves examining the properties of the eigenvalues to determine a "good" bandwidth *a priori.* A "good" bandwidth is one for which the condition number of T is small. Next the matrix is reduced to banded form, A_b, using a series of operations to eliminate off diagonal elements. The operations are accumulated in a matrix T. If T is found to be ill-conditioned, then the tolerance for Step 3 is tighned and Steps 3 and 4 are repeated. Finally, the matrix A_b is brought to complex, upper-triangular, banded form using a series of Givens transformations. We note that the transformations used in Step 4 are scaled to provide reduction in their condition numbers.

An Illustration

The figure shown illustrates the banding algorithm. The first operation shows the effect of bringing the system to Schur form. After the matrix has been brought to Schur form, the matrix is analyzed to determine a "good" bandwidth. Here we choose a bandwidth of 2. The second set of operations shows how the algorithm reduces a diagonal of the matrix. The third set of operations shows how the remaining diagonals are eliminated to produce the final upper-triangular, banded matrix.

Test Case

The algorithm described has been coded into Fortran and installed into our Pro-Matlab implementation using the Pro-Matlab MEX facility. We chose as a test set a set of single input, single output systems with state order ranging from 20 to 80. Matrix coefficients were generated from a random number generator. For each case we computed 200 frequency points. The table shows times for the Pro-Matlab **bode** function versus times for our **bodeq** function. As one can see, the new algorithm reduced the computation time from 75 to 88 percent.

Extensions and Future Work

The algorithm has also been applied to time simulation of linear, time-invariant systems. The banding strategy and algorithm could be extended to generalized state space systems. In this case, we would band the A and E matrices simultaneously. Another possible area of future work would be production of better banding algorithms. The current algorithm is bases on solution of Sylvester equations and has a limitation: the algorithm cannot band systems well when all eigenvalues are very closely spaced. It should be possible to band these matrices using different algorithms.

Summary

In summary, we have developed a new algorithm for computing frequency responses of state space models. In this development, we have taken into account the two prime computational issues: efficiency and sensitivity. We showed that the algorithms worked on a test problem and was able to reduce computational time considerable without a notable cost in accuracy. Finally, we proposed that the banding strategy may provide further application in control system design.

JPL

A Fast, Reliable Algorithm for Computing Frequency Responses of State Space Models

Matt Wette
Jet Propulsion Laboratory, Caltech

JPL

- Introduction to the Problem
- Typical Approach
- Computational Issues
- A New Banding Algorithm
- Test Case
- Extensions and Future Work
- Summary

Introduction to the Problem

Consider a transfer function $G(j\omega)$ with state space realization given by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

where $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^n$, and $y(t) \in \mathbb{R}^p$.

$G(j\omega)$ is associated with $\{A, B, C, D\}$ through

$$G(j\omega) = C(j\omega I - A)^{-1}B + D$$

Problem: desire $G(j\omega_k)$ for many (hundreds) of values of $j\omega_k$
where $n < 200$ (medium order systems).

Typical Approach

1 Transform the system realization:

$$\{A, B, C, D\} \xrightarrow{T} \{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}\} := \{T^{-1}AT, T^{-1}B, CT, D\}$$

2 For each ω_k do
  a) solve $(j\omega_k I - \tilde{A})X = \tilde{B}$ for X
  b) compute $G(j\omega_k) = \tilde{C}X + \tilde{D}$

Compuatational Issues

- Solution of $(j\omega_k I - \tilde{A})X = \tilde{B}$ must be efficient.
  This is usually the limiting factor.
- Given the presence of finite precision arithmetic
  $\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}\}$ must be an accurate realization of $G(j\omega)$.

Sensitivity of the Transformation

Consider

$$\{A+\Delta A, B+\Delta B, C+\Delta C, D+\Delta D\} \xrightarrow{T} \{\tilde{A}+\Delta\tilde{A}, \tilde{B}+\Delta\tilde{B}, \tilde{C}+\Delta\tilde{C}, D+\Delta D\}$$

Then we have

$$\kappa(T)^{-2}\frac{\|\Delta A\|}{\|A\|} \leq \frac{\|\Delta\tilde{A}\|}{\|\tilde{A}\|} \leq \kappa(T)^{2}\frac{\|\Delta A\|}{\|A\|}$$

$$\kappa(T)^{-1}\frac{\|\Delta B\|}{\|B\|} \leq \frac{\|\Delta\tilde{B}\|}{\|\tilde{B}\|} \leq \kappa(T)\frac{\|\Delta B\|}{\|B\|}$$

$$\kappa(T)^{-1}\frac{\|\Delta C\|}{\|C\|} \leq \frac{\|\Delta\tilde{C}\|}{\|\tilde{C}\|} \leq \kappa(T)\frac{\|\Delta C\|}{\|C\|}$$

where $\kappa(T) := \|T\|\|T^{-1}\|$ and $\|\cdot\|$ is some consistent norm.

Efficient Solutions to $(j\omega_k I - \tilde{A})X = \tilde{B}$

- A a general matrix $\Rightarrow \mathcal{O}(n^3)$ flops/ω_k
- $\tilde{A}$ an upper Hessenberg or Schur matrix $\Rightarrow \mathcal{O}(n^2)$ flops/ω_k
- $\tilde{A}$ a matrix in diagonal form $\Rightarrow \mathcal{O}(n)$ flops/ω_k
- $\tilde{A}$ a matrix in Jordan canonical form $\Rightarrow \mathcal{O}(n)$ flops/ω_k
- $\tilde{A}$ a block-diagonal matrix $\Rightarrow \mathcal{O}(n^l)$ flops/ω_k, where $1 \leq l \leq 2$
- $\tilde{A}$ a *banded matrix* $\Rightarrow \mathcal{O}(n \cdot bw)$ flops/ω_k where bw is the *bandwidth*, $0 < bw < n$

A New Banding Algorithm

Given a full general A matrix, produce A_b where A_b is upper triangular and banded. The algorithm is essentially

1 Reduce A to upper real Schur form, A_s

2 Order the eigenvalues appearing on diagonal blocks of A_s to produce A_o

3 Analyze A_o to determine a "good" bandwidth (to make $\kappa(T)$ small)

4 Uses an algorithm based on solving Sylvester equations to band A_o, producing A_b

5 Convert the quasi-upper triangular matrix A_b to complex, upper triangular form.

Step 4 uses transformations of the form

$$T_{i,j}=\begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & x_{i,j} & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}\begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & d_4 & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}$$

An Illustration

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_s} \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \otimes & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{T_{1,3}} \begin{bmatrix} \times & \times & 0 & \times & \times \\ 0 & \times & \times & \otimes & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{T_{2,4},T_{3,5}} \begin{bmatrix} \times & \times & 0 & \times & \times \\ 0 & \times & \times & 0 & \times \\ 0 & 0 & \times & \times & \otimes \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & 0 & \otimes & \times \\ 0 & \times & \times & 0 & \otimes \\ 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{T_2} \begin{bmatrix} \times & \times & 0 & 0 & \otimes \\ 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{T_3} \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

Test Case

Single-Input, Single-Output Model with Random Coefficients
Pro-Matlab **bode** function versus banded **bodeq** function

pts.	n	bode (sec)	bodeq (sec)	bw	$\kappa(T)$
200	20	39.9	4.8	3	250
200	30	81.1	10.6	3	350
200	40	139.8	24.0	13	45
200	50	211.7	41.6	17	38
200	60	300.6	64.8	20	66
200	70	407.7	104.6	31	27
200	80	527.1	135.1	32	67

Reduction in time from 75% to 88%

Extensions and Future Work

- Application to time simulation of $\{A, B, C, D\}$
- Application to extended (or generalized) models:

$$E\dot{x} = Ax + Bu\,, \quad y = Cx + Du$$

- Better banding algorithms: Banding strategy has good potential but current algorithm has some limitations.

Summary

- Developed new algorithm for computing frequency responses of state space systems.
- The algorithm provides a method for trading off the two computational issues at hand: sensitivity and efficiency.
- The algorithm was shown to provide large saving in computational time on a set of test problems.
- The strategy has some potential for other applications on medium order models.