Proceedings of the 39th IEEE
Conference on Decision and Control
Sydney, Australia • December, 2000

Riccati Equation-based Stabilization of Large Scale Dynamical Systems

X. Rao and K. A. Gallivan
Florida State University
Tallahassee FL, 32306
{gallivan}@cs.fsu.edu

P. Van Dooren
CESAME – Université catholique de Louvain
Av. G. Lemaître, 4, B-1348 Louvain-la-Neuve, Belgium
vdooren@csam.ucl.ac.be

Abstract

In this paper we discuss the stabilization of large scale linear time invariant dynamical systems via feedback. Efficient schemes based on the Discrete Riccati Difference Equation are presented.

1 Introduction

In this paper, we focus on the stabilization of a discrete-time system

$$x_{i+1} = Ax_i + Bu_i, \tag{1}$$

where A and B are $n \times n$ and $n \times p$ real matrices which are known, and x_i and u_i are vectors of dimension n and p respectively. The stabilization of the system requires the computation of a $p \times n$ feedback matrix F such that all eigenvalues of $A - BF$ are inside the unit circle and therefore the system defined by replacing A with $A - BF$ is stable. For small and moderate values of n, F can be computed via pole placement or the solution of a matrix equation, e.g., Riccati or Lyapunov equations. The computational requirements for standard algorithms for these approaches, however, is prohibitive for large values of n. Fortunately, when n is large and $p << n$, the system matrix A and/or input matrix B are typically very sparse. Algorithms for such problems must therefore exploit this structure in order to efficiently compute a stabilizing feedback.

2 Saad's Approach

A major contribution to solving large scale stabilization problems with a few unstable eigenvalues is Y. Saad's projection method [1]. In this algorithm, stabilization or eigenvalue assignment is only imposed on a small invariant subspace that contains the unstable invariant subspace of A. Such an approach is often effective, but it can have convergence difficulties and the need for a basis of the invariant subspace can cause excess space requirements for very large systems. In this paper, we discuss efficient alternatives that address the convergence difficulties. Details on all of the algorithms can be found in [2].

In Saad's projection algorithm, a left invariant subspace V' of A (with presumably small dimension), that contains the left unstable invariant subspace of A is computed. There are two major classes of methods that can be used. The first computes the unstable eigenvalues and recovers their eigenvectors by some form of inverse iteration. The second class computes the basis directly by subspace iteration-like methods. The low-order projected system $(V'AV, V'B)$ is then stabilized and the reduced feedback F_u is lifted back to form a stabilizing feedback $F = F_uV'$ of the original system (A, B).

Methods in the first class benefit from years of sparse eigenvalue algorithm research but often require very high accuracy in the eigenvalues in order to produce the basis and hence result in more computation than necessary for stabilization.

Effective convergence is one of the main issues of the second class of methods. The convergence of subspace iteration-like (SSI) methods which generate the sequence of approximations to the invariant subspace starting from initial subspace V_0 and updating V_i by extracting an orthogonal basis of $A'V_{i-1}$ is usually consistent with the separation between desired eigenvalues and undesired eigenvalues in absolute value. In practice, it is often difficult to tune the parameters of such methods to converge even this quickly. They can be accelerated and some parameter sensitivity mitigated by the use of Stewart's SRR (Schur-Rayleigh-Ritz) refinement [5]. The acceleration is achieved by enlarging the size of initial subspace V_0, extracting the Schur vectors U_i corresponding to largest (or unstable) eigenvalues of $V_i'A'V_i$ and combining V_iU_i as the basis of the approximated invariant subspace. In [2], we have investigated a version of SSI/SRR that applies these ideas. It is this algorithm that is used in the comparisons below.

A second source of difficulty for Saad's method is that it is, by definition, a two-phase process: find the basis then stabilize. Experience and empirical testing shows that a stabilizing feedback can often be found

0-7803-6638-7/00$10.00 © 2000 IEEE

with approximations available long before the eigensolver would have any confidence in the basis of the unstable space. Simply computing a feedback on every iteration of the eigensolver is too expensive so a more seamless method of integrating the feedback computation with the update of the basis is needed. Finally, the major drawback with this approach is the need for a basis of the invariant subspace. Storage problems for large dynamical systems can result, therefore, it is worthwhile to look for methods that do not require the basis. We have developed a family of methods starting from the Discrete Riccati Difference Equation that addresses these concerns, is competitive with Saad's method when Saad's method does well and is successful for many problems where Saad's method fails.

3 Discrete Riccati Equation Stabilization

The major results of this paper are based on the discrete-time Riccati equation (DRE)

$$P = A'(P - PB(R + B'PB)^{-1}B'P)A + Q \qquad (2)$$

where R and Q are $p \times p$ and $n \times n$ non-negative matrices and Q is usually decomposed into CC'. The methods in this paper solve the DRE via two basic iterative approaches. The first is the discrete-time Riccati difference equation (DRDE)

$$P_{i+1} = A'(P_i - P_iB(R + B'P_iB)^{-1}B'P_i)A + Q. \qquad (3)$$

The DRDE is a *fixed point* iteration method to solve the DRE and converges linearly with the rate $\rho^2(A - BF_\infty)$. The second basic approach taken is motivated by Newton's method which, in theory, has quadratic convergence when used to solve non-linear equations.

The most general results about DRE and DRDE convergence are given in [3]. It is shown there that under the condition of stabilizability of (A, B), a stabilizer and non-negative solution P of DRE (2) exist and a stabilizing feedback F can be computed by $(R + B'PB)^{-1}B'PA$. Whether the solution of DRDE (3) converges to the stabilizing solution of DRE depends on properties of (A', C) and the initial condition P_0.

A discussion of the solution and convergence of Newton's method for the DRE can be found in [6] and its references. Rao merges several results from [6] and summarizes sufficient conditions for the existence of a stabilizing solution, convergence and the convergence rate of Newton's method applied to the DRE in a unifying theorem [2].

4 The SQR Stabilization Algorithm

For the purpose of stabilization, we have freedom in choosing R, C and P_0. Our stabilization algorithms using various combinations of parameter settings for C (or Q) and P_0. In order to exploit a relatively low dimensional unstable space as Saad assumes, we take $Q = 0$ from which it follows that the rank of P_i will be non-increasing. P_i can be represented in a compact fashion via a basis for the current approximation to the unstable space by using one of the square root iterations developed in the literature for several scenarios, including the DRDE.

A special case of the square root form of DRDE, introduced in [4] for Kalman filtering, the SQR stabilization algorithm has the form

$$\begin{bmatrix} R^{1/2} & B'P_i^{1/2} \\ 0 & A'P_i^{1/2} \end{bmatrix} U_i = \begin{bmatrix} (R_i^\epsilon)^{1/2} & 0 \\ \tilde{K}_i & P_{i+1}^{1/2} \end{bmatrix} \qquad (4)$$

where U_i is orthogonal and the dimension of $P_i^{1/2}$ is $n \times l$, the same as $P_0^{1/2}$. Note that the QR decomposition is computed for a small matrix with size $(p+l) \times p$ (the first row of (4)) and feedback F_i can be computed from $(R_i^\epsilon)^{1/2}$ and $\tilde{K}_i$.

The feedback generated in the limit moves the unstable eigenvalues of A, λ to their unit circle mirror images, $1/\lambda$, and leaves the stable eigenvalues unchanged. The SQR iteration can produce the same sequence of subspaces as orthogonal subspace iteration (SSI) with only an additional economical QR decomposition of $P_i^{1/2}$ since the updating of $P_i^{1/2}$ has the form $P_{i+1}^{1/2} = A'P_i^{1/2}U_i^{22}$. If $P_0^{1/2}$ is taken to be the same initial subspace basis as used for a convergent SSI, SQR will converge. Even if SSI does not converge, SQR will converge under conditions related to the ability of the Stewart's Schur-Rayleigh-Ritz refinement (SRR), [5], to extract a convergent subsequence of approximated unstable invariant subspaces (see [2] for a detailed discussion of the SSI/SRR used in performance comparisons with SQR).

Consideration of the relationship to SSI also yields insight into the significant difference between the methods that is responsible for the superiority of the SQR approach. While SSI and SQR produce the same space, they do not produce the same basis for that space. The update to the basis $P_{i+1}^{1/2} = A'P_i^{1/2}R$ is used in SSI. The application of A' emphasizes the unstable directions in the current approximation. The postmultiplication with R simply orthogonalizes the new basis. In SQR, however, R is replaced by U_i^{22} – a submatrix of U_i. This transformation does not orthogonalize but it does tend to damp the stable directions in the current basis. So SQR is a true two-sided process, A' emphasizing the unstable directions in which we are interested and U_i^{22} damping the stable directions in which we are not. The literature does not address the issue of the convergence to a stabilizer of the DRDE with $Q = 0$. In [2] a argument is made that the above description heuristically characterizes the behavior of SQR but a rigorous proof is not given. In fact the following can be shown and is the subject of a forthcoming paper:

Theorem 1 *If $Q = 0$ and $P_0^{1/2}$ has components in all unstable directions, with no poles on the unit circle, and (A, B) is stabilizable, then SQR converges to a stabilizer, F, such that the stable poles do not move and the unstable poles move to their reciprocals.*

Convergence is not enough for an effective algorithm of course. A reliable termination check is needed in order to avoid iterating longer than necessary. The test occasionally extracts an approximation to the unstable subspace V_i via SRR from $P_i^{1/2}$ and compares the eigenvalues of $V_i'(A - BF_i)V_i$ and unit circle mirror images of eigenvalues of $V_i'AV_i$. Empirical evidence indicates that this test effectively detects stabilization much earlier than other convergence tests for square root-like methods.

The combination of a more sophisticated termination criterion with the SQR algorithm produces a stabilization algorithm that often converges much faster than careful implementations of Saad's approach using orthogonal subspace iteration to determine the invariant subspace basis. SQR also tends to be much more robust in terms of parameter selection.

5 The TSQR Stabilization Algorithm

In order to improve the linear convergence of the SQR algorithm, we have explored the use of a variant of Newton's method to solve the DRE for stabilization. Let $\mathcal{R}(X)$ be the residual of the DRE with X as an approximate solution:

$$\mathcal{R}(X) = -X + A'XA - A'XB(R + B'XB)^{-1}B'XA + Q.$$

Applying Frechet differentiation and the definition of Newton's method yields a matrix form of Newton's iteration for the DRE ([6]):

$$X_i - A_{i-1}'X_iA_{i-1} = Q + F_{i-1}'RF_{i-1}, i = 1, \cdots, \quad (5)$$

where

$$F_i = (R + B'X_iB)^{-1}B'X_iA, \; A_i = A - BF_i \quad (6)$$

As mentioned above the convergence of this algorithm is discussed in [2] but even as stated there are difficulties with its direct implementation. It is sensitive to the initial guess; as stated it requires a stabilizing initial feedback (which is the whole point of the problem); the discrete Lyapunov equation solved during each iteration does not have a bounded solution for unstable A_i; the cost to solve the Lyapunov equation is almost the same as solving the DRDE; and the positive semidefiniteness must be maintained in the presence of numerical errors. An alternative and approximate implementation of Newton's method must be developed to address these problems.

The truncated square root Newton's iteration (TSQR) is defined as follows:

Algorithm 1 Truncated Square Root Newton's Iteration: *Let A be $n \times n$ and B be $n \times p$.*

1. *Specify integers $l > 0$ and m (m is zero if we choose $Q = 0$). Take $(X_0)^{1/2}$ and $Q^{1/2}$ as random matrices with dimension $n \times l$ and $n \times m$. Take $R = I$ and $F_0 = (R + B'X_0B)^{-1}B'X_0A$.*

2. *Specify a small integer $K > 0$ and an integer N. Let $(X_0^K)^{1/2} = (X_0)^{1/2}$. For i from 1 to N,*

   (a) *For j from 1 to K, do*

      i. $(\tilde{X}_i^j)^{1/2} = \left[(A - BF_{i-1})'X_i^{j-1} \;\; Q^{1/2} \;\; F_{i-1}'R^{1/2}\right].$

      ii. *Apply economic SVD on $(\tilde{X}_i^j)^{1/2}$ to get $(\tilde{X}_i^j)^{1/2} = U_i^jS_i^jV_i^j$.*

      iii. *Update $(X_i^j)^{1/2} = U_i^j(:, 1:l)S_i^j(1:l, 1:l)$.*

   (b) *Compute $F_i = (R + B'X_i^jB)^{-1}B'X_i^jA$ and let $(X_{i+1}^0)^{1/2} = (X_i^K)^{1/2}$.*

   (c) *Compute some stabilization criterion or convergence criterion. Stop if any criterion is reached.*

Note we use X_{i-1}^K instead of 0 as the initial guess of the next iteration's approximation of Lyapunov equation. When applied to stabilize the system (A, B), the *full rank* implementation of the algorithm stabilizes and converges much faster than the DRDE-based fixed point algorithms such as SQR and the CSQR algorithm discussed later.

For large scale systems, especially large sparse systems, a full dimension implementation with an $n \times n$ X_i^j is impractical. Unless $Q = 0$, where the rank of the solution P or X_+ is at most the number of unstable eigenvalues, we have to use some form of truncation to approximate $(X_i^j)^{1/2}$ such that number of columns of the approximation $(X_i^j)^{1/2}$ is acceptably small. In the form above, we only keep the dominant part of X_i^j as measured by its largest l singular values and vectors. In [7], Verlaan uses the same idea to decompose the DRDE into a square root form. However, as discussed in [2], the methods differ based on the form of the DRDE upon which they operate. The square root algorithm and truncation algorithm described in this paper is, we believe, easier to understand and to implement.

The TSQR algorithm has a wealth of parameters and truncation choices that subsume algorithms such as SQR and provides a setting for more general algorithm investigation. For $Q = 0$, $K = 1$, and $(X_0)^{1/2}$ low-rank the method is essentially SQR. Increasing K in this case can provide a significant improvement over SQR for certain problems. With nonzero Q, TSQR is very sensitive to the choice of l and as a result for poor choices K may become large implying a substantial increase in computation. In [2], a technique to improve this situation by introducing an SSI iteration into the TSQR iteration at two key points.

If used carefully, TSQR can be used to approximate the DRDE for large scale optimal filtering problems where all the parameters are predetermined. For stabilization purposes, TSQR can find a stabilizing feedback efficiently by manipulating parameters Q, R and the initial guess with more freedom than the SQR and C-SQR algorithms. However, further work is required on automatically selecting the particular version of TSQR appropriate for a given problem.

6 The CSQR Stabilization Algorithm

SQR and TSQR address the first two difficulties with Saad's method. However, since l, the the rank of the (T)SQR approximation, or the rank of $P_0^{1/2}$, is at least the number of unstable eigenvalues of A, both algorithms still require the propagation of a basis of the unstable invariant subspace.

6.1 The Algorithm

A true low-rank stabilization algorithm that does not require the propagation of an estimate of the basis of the unstable space can be developed from the observation that if $P_0 = 0$ in the DRDE, $P_{i+1} - P_i$ is non-negative and its rank is non-increasing. The use of $P_{i+1} - P_i$ leads to the use of another well-known recurrence called the square root Chandrasekhar [4] algorithm (denoted CSQR).

Let L_i be the square root of $P_{i+1} - P_i$, starting from $L_0 = C$, $R_0^\epsilon = R$ and $\tilde{K}_0 = 0$. CSQR has the form

$$\begin{pmatrix} (R_{i-1}^\epsilon)^{1/2} & B'L_{i-1} \\ \tilde{K}_{i-1} & A'L_{i-1} \end{pmatrix} U_i = \begin{pmatrix} (R_i^\epsilon)^{1/2} & 0 \\ \tilde{K}_i & L_i \end{pmatrix} \quad (7)$$

where U_i is orthogonal and the feedback F_i can be computed via $\tilde{K}_i(R_i^\epsilon)^{-1/2}$. The dominant computation in each iteration of CSQR is $A'L_{i-1}$, whose rank is just the rank of C. CSQR will converge for any choice of C. For stabilization however, C should satisfy the condition that (A', C) is stabilizable. Typically, C can be taken as a matrix of rank 1, but, in general, its rank must be at least the largest geometric multiplicity of any unstable eigenvalue of A. Note that the dimensions of the matrices propagated do not depend on the dimension of the unstable space. The CSQR algorithm therefore addresses all three difficulties with Saad's method and in the form above is practical for many problems.

6.2 Acceleration

The convergence of CSQR is more sensitive to tuning than SQR. In [2], techniques that improve its performance significantly are discussed in detail. These include scaling the system, preprocessing C, and restarting. Also, the stabilized spectrum may move the original stable eigenvalues unlike Saad's method or SQR. We have found however that in practice, with care, they do not move that much.

When using the DRDE, large eigenvalues tend to be stabilized or moved close to the unit circle very quickly but, the unstable eigenvalues that are close to the unit circle need many more iterations to stabilize. With $\rho < 1$, when using CSQR on $(A/\rho, B/\rho)$, A/ρ will enlarge any unstable eigenvalue λ of A to λ/ρ which will be stabilized or at least be moved close to the unit circle very quickly by $A/\rho - (B/\rho)F_i$, and λ will be stabilized by $A - BF_i$.

This scaling technique works very well if there exists a $\rho < 1$ such that the number of eigenvalues of A with absolute value not less than ρ is very small. A special case of this is when some unstable eigenvalues of A are very close to the unit circle and stable eigenvalues of A are well-separated from the unit circle (with the largest stable eigenvalues having modulus near ρ). For such a special case, the stabilization and feedback convergence with general CSQR are very slow, but scaling A accelerates both stabilization and feedback convergence.

In some cases, however, there may not be such a simple separation between the unstable eigenvalues around the unit circle and all of the stable eigenvalues. Some stable eigenvalues may be near the unit circle as well and therefore for the scaled problem will appear in the unstable set. It can be shown that any unstable eigenvalue of A remains stabilizable in A/ρ, but a stable eigenvalue of A may not remain stabilizable for A/ρ. The action of CSQR on unstabilizable unstable eigenvalues of A/ρ is critical to this acceleration technique.

This problem is considered in detail in [2]. It is shown there that if a stable eigenvalue becomes unstabilizable in the scaled system, the DRDE iteration will produce eventually a divergent P_i. However, since the unstabilizable eigenvalue must be dominated in magnitude by the stabilizable unstable eigenvalues (those that are unstable in the original system) the feedback F_i converges to a stabilizer before P_i diverges and in practice the problem can be solved.

Preprocessing C to get the initial value, L_0, also improves performance. This preprocessing takes the simple form of an SSI-like iteration, e.g., $L_0 = (A')^k C$. This emphasizes the unstable directions over the stable ones allowing CSQR to concentrate on them first. Amazingly, a small investment of $K = 5$ or 10 tends to result in a significant reduction of the number of iterations required for stabilization. Investing more can be counterproductive since it then tends to emphasize a subset of the unstable directions requiring the subsequent use of a deflation procedure to address the remaining unstable eigenvalues.

Finally, restarting has proven an effective acceleration method for SQR. A new L_0 is chosen occasionally. The iteration is restarted with $R_0^\epsilon = R$, $K_0 = 0$ and $A \leftarrow A - BF_i$. Of course, A is not explicitly updated. The action of A on a vector is computed from the original matrix and all of the F_j that have resulted from restarting. In the experiments below $L_0^{new} = L_i$ at fixed intervals in the iteration. More sophisticated monitoring schemes are under investigation.

6.3 Experiments

In [2] experimental evidence is presented demonstrating the strengths and weaknesses of each of the methods. SQR and TSQR can be highly effective. Below we summarize two experiments that show CSQR is also highly promising.

The first experiment considers a random 100×100 matrix with 4 unstable eigenvalues and many stable and unstable near the unit circle. Such a system causes difficulties for SSI and SQR. Table 1 shows the rank used, the cost of the iterations until stabilization in terms of the number of matrix vector products, and, in the same terms, the cost of the iterations until detection of stabilization. PCSQR, RCSQR, and SCSQR refer to preprocessed, restarted and scaled CSQR respectively. The latter two also use preprocessing. Note the significant performance improvement over SQR and SSI. The need for further work on termination criteria for CSQR is also clear. Due to the form of the algorithm the termination criterion used for SQR is not applicable to CSQR. A simple estimation strategy for the spectral radius of $A - BF_i$ is used.

For the second experiment, A is a 100×100 matrix with the spectrum shown in Figure 1 (48 unstable poles). B is a 100×20 random matrix. A multiple-input system is required to keep the stabilization conditioning accept-

Table 1: Cost for Stabilization for Experiment 1.

Algorithm	Rank	Cost to Stable	Cost to Detected
SSI	4	> 1600	> 1600
SQR	4	1200	> 1600
PCSQR	1	310	410
RCSQR	1	250	350
SCSQR	1	80	150

able, i.e., stabilizing large numbers of eigenvalues with a single-input is very difficult. C is a random vector generated by RAND. SCSQR with rank 1 and scaling of A by 0.9 stabilized the system within 65 iterations or less than two iterations per unstable eigenvalue. The stabilization was detected by 100 iterations and the feedback difference declined steadily. SSI, SQR or TSQR would require the propagation of a basis with at least 48 vectors while CSQR updates a single vector while stabilizing a large number of unstable eigenvalues.

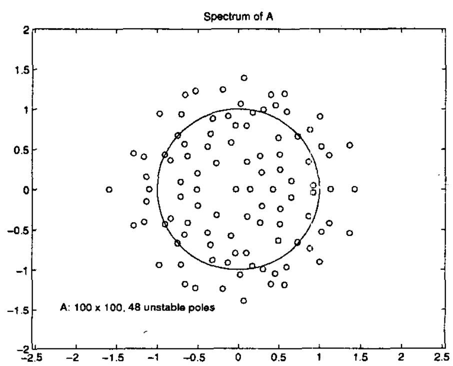

Figure 1: Spectrum of A, Experiment 2

7 Conclusions

In general, the results here and in [2] indicate that the proposed methods have tremendous promise to efficiently stabilize large scale systems.

SQR merges the two phases of Saad's method successfully and performs as good as or better than subspace methods while yielding the same stable eigenvalue distribution after feedback and with complexity comparable to subspace methods. The proposed SQR termination criterion detects stabilization much earlier than previously used techniques thereby improving efficiency. SQR's convergence is improved and its relationship to other methods is demonstrated by the TSQR method. The general algorithm space defined by TSQR deserves more attention and the use of a CSQR-like square root strategy with an approximate Newton's method has yet to be explored.

CSQR is very efficient in time and space and has multiple effective convergence accelerators. Its operation is more flexible than SQR. It is, however, more complicated than SQR to tune and the closed-loop spectrum is not as simply related to the original spectrum as that of SQR and SSI.

There are several directions for current and future work. The merger of the two phases may be adaptable to other eigenvalue methods for use in stabilization, i.e., those more sophisticated than SSI. The early stabilization behavior of the DRDE requires more careful analysis via a mix of convergence theory for the eigenvalues, eigenspaces, feedback, and the DRDE. More effective automatic restarting/scaling strategies are required for CSQR as well as a theoretical understanding of the effect of restarting. While SQR has an effective termination criterion, that of CSQR needs further consideration. The control of the final positions of the eigenvalues for CSQR also requires better understanding. Initial attempts at applying these methods to continuous time problems are made in [2] and are encouraging, but a substantial amount of improvement appears possible.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. CCR-97-96315 and by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture.

References

[1] Y. Saad: *Projection and deflation methods for partial pole assignment in linear state feedback* IEEE AC, Vol. 33, No. 3(1988)

[2] X. Rao: *Large scale stabilization with linear feedback* M.S. Thesis, Department of Computer Science, Florida State University, 1999.

[3] C. E. De Souza, M. R. Gevers and G. C. Goodwin: *Riccati Equations in optimal filtering of non-stabilizable systems having singular transition matrices*, IEEE AC, Vol. 31, No. 9(1986)

[4] M. Morf and T. Kailath: *Square-Root Algorithms for Least-Squares Estimation*, IEEE AC, Vol. 20, No. 4(1975)

[5] G. W. Stewart: *Simultaneous iteration for computing invariant subspace of non-Hermitian matrices* Numer. Math., 25, 123-136 (1976)

[6] C-H. Guo: *Newton's method for discrete algebraic Riccati equations when the closed-loop matrix has eigenvalues on the unit circle*, SIAM J. Matrix Anal. Appl., 20(2), 1999.

[7] M. Verlaan, *Efficient Kalman Filtering Algorithms for Hydrodynamic Models*, Ph.D. thesis, Delft University of Technology, 1998.