Stabilization of Large Scale Dynamical Systems

X. Rao and K. A. Gallivan
Florida State University
Tallahassee FL, 32306
{rao,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

Keywords: stabilization, large scale, dynamical systems, linear, time-invariant

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. The SQR variant is described in detail and a more efficient version, CSQR, is motivated.

1 The Problem

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. We will also motivate and algorithm that avoids the need for an explicitly formed basis of the invariant subspace. The latter will be explored in detail in a forthcoming paper. Details on all of the algorithms can be found in [6].

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 $F = F_u V'$ 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 [4]. 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 [6], 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 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) and the discrete-time Riccati difference equation (DRDE)

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

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

where R and Q are $p \times p$ and $n \times n$ non-negative matrices and Q is usually decomposed into CC'. 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.

For the purpose of stabilization, we have freedom in choosing R, C and P_0. We have developed low-rank stabilization algorithms using various combinations of parameter settings for C (or Q) and P_0. Starting from non-negative P_0, P_i in the DRDE will keep the non-negative property. If Q is chosen as zero, the rank of P_i will be non-increasing. If $P_0 = 0$, $P_{i+1} - P_i$ will be non-negative and its rank will be non-increasing. The low-rank stabilization algorithms in this paper are based on these non-negative and non-increasing rank properties and square-root decomposition of these non-negative matrices (see [6] for the derivations of these facts).

4 The SQR Stabilization Algorithm

Square root forms of iterations like the DRDE have been developed for several scenarios in the literature. The square root algorithm (SQR) of this paper is based on the DRDE with $Q = 0$. 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. As a special case of the square root form of DRDE, introduced in [2] 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 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 SQR iteration can produce the same sequence of subspaces as 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 SSI, SQR will converge. Even if SSI does not converge, SQR will converge under conditions related to the ability of the SRR to extract a convergent subsequence of approximated unstable invariant subspaces.

The use of an effective termination check is needed in order to avoid iterating longer than necessary. The square root algorithms in the literature usually use a stopping criterion analogous to the convergence of the feedback increment $F_i - F_{i-1}$, which is determined by smallest distance of eigenvalues (especially unstable eigenvalues) of A to the unit circle and would suffer slow convergence if some eigenvalues are too close to the unit circle. We have developed and evaluated a more effective method which can catch stability of $A - BF_i$ earlier in the SQR iteration. This is achieved by extracting an approximation to the unstable subspace V_i via SRR from $P_i^{1/2}$ and comparing 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, referred to as the ΔV test below, effectively detects stabilization much earlier than other convergence tests for square root-like methods.

5 Experiments and Scaling

The following examples investigate the efficiency of SQR when applied to a system that is well-conditioned and easy to stabilize and another system that is less well-conditioned and hard to stabilize.

Example 1 *The system matrix A is constructed by randomly generating a 100×100 matrix with MATLAB RANDN and scaling it to $\tilde{A}$ so that the spectral radius of $\tilde{A}$ is 0.9.*

The 100×1 matrix B is generated randomly with RAND as is a 1×100 matrix $\tilde{F}$. Construct $A = \tilde{A} + B\tilde{F}$. All eigenvalues of A (dots in Figure 1) are well-separated from the unit circle and only two are unstable. The norm $\|\tilde{F}\|_2 = 9.9734$ and eigen-condition number of $\tilde{A}$ is $\kappa_2(X, \tilde{A}) = 110.8009$. So the system (A, B) should be well-conditioned and easy to stabilize. Figure 3 and Figure 4 show the results of rank 2 SQR. Figure 5 and Figure 6 show the result of rank 3 SQR. $P_0^{1/2}$ is randomly generated with RAND. The spectral radius of $A - BF_i$ from both rank 2 and rank 3 SQR converges within 7 iterations, which is shown in Figure 3 and Figure 5. With the new stabilization criterion ΔV_i related to projections of $A - BF_i$ and A on V_i, Figure 2 catches stabilization of both Rank 2 and Rank 3 SQR at step 7 with tolerance 10^{-2}, while residuals from SSI and $\|F_i - F_{i-1}\|_2$

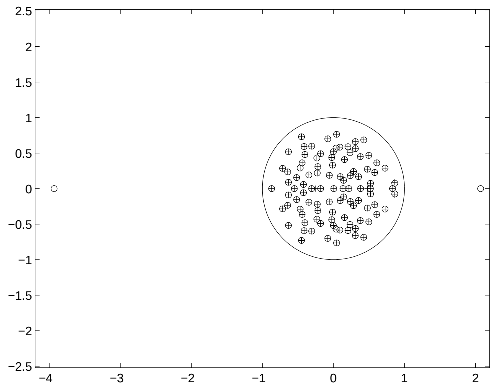

Figure 1: SQR: Spectrums for Example 1

have not reached 10^{-2} (see Figure 4 and Figure 6). Furthermore, Figure 1 shows the spectrum of $A - BF_5$ (the $+$ symbols) converges to a stable configuration in 5 iterations. We also see that the only eigenvalues moved are the two unstable ones of A. Figure 4 and Figure 6 show the convergence of the feedback and V_i with SQR, SSI convergence and another stabilization criterion $(A - BF_i) \cdots (A - BF_1)B$. In Figure 4, the ranks of SQR and SSI are taken to be the order of the unstable space and the subspace sequence (orthogonal basis) from $P_i^{1/2}$ converges to an unstable invariant subspace of A'. We see feedback of SQR and the SSI residual converge with the same rate. In Figure 6, we use the wrong rank for the subspace, and SSI has trouble converging and needs Stewart's refinement with significant additional computational cost. The feedback of SQR still converges with a rate consistent to closed-loop spectral radius as predicted.

Example 2 *In this example, A is a 100×100 matrix generated by RANDN of MATLAB and is scaled so that A has the spectrum described in Figure 7. B is a 100×1 matrix generated by RAND of MATLAB. Figure 7 shows there are plenty of eigenvalues, both stable and unstable, which are very close to the unit circle. It is, therefore, expected that stabilization via SQR or SSI will be very costly.*

Figure 8 to Figure 12 show the efficiency of SSI and SQR of rank 4 (correct order) and rank 5 (incorrect order). Figure 9 and Figure 11 show about 300 steps of SQR are taken before the closed-loop spectral radius of $A - BF_i$ starts to converge. With ΔV_i, Figure 8 catches stability of $A - BF_i$ at about step 350 for Rank 4 SQR and step 280 for rank 5 SQR with tolerance 10^{-2}, while both $\|F_i - F_{i-1}\|_2$ from SQR, subspace residuals for unstable subspace V_i and SSI reach accuracy 10^{-2} to 10^{-4} and hence do not indicate convergence. Figure 7 shows that the closed-loop spectrum of $A - BF_i$ converges at step 300 of SQR and only unstable eigenvalues of A are moved.

Example 1 illustrates that for a well-conditioned stabilization problem where A has only few unstable eigenvalues and all eigenvalues of A are well-separated from the unit circle, SQR is very efficient and stabilization is reached within only

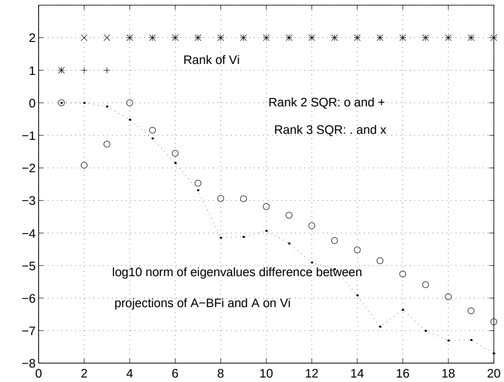

Figure 2: SQR: $\log_{10}(\Delta V_i)$, Example 1

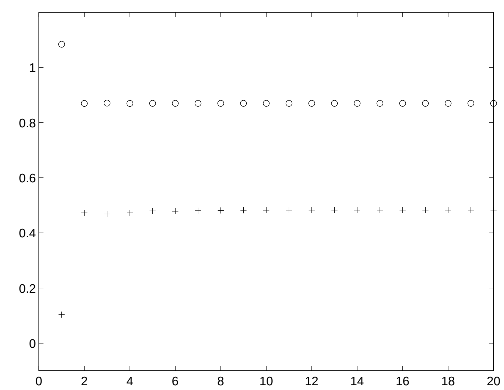

Figure 3: SQR: $A - BF_i$ (o) and $V_i'(A - BF_i)V_i$ (+), Example 1

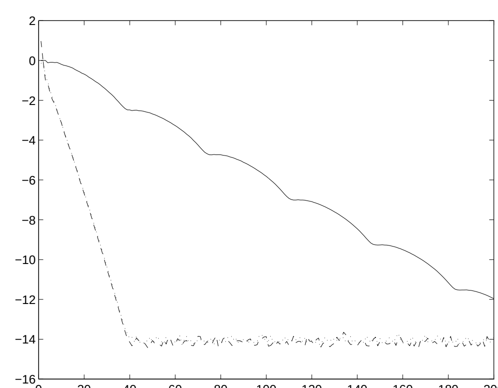

Figure 4: SQR: Feedback (- -), SSI (.) and $\prod(A - BF_i)B$ (–), Example 1

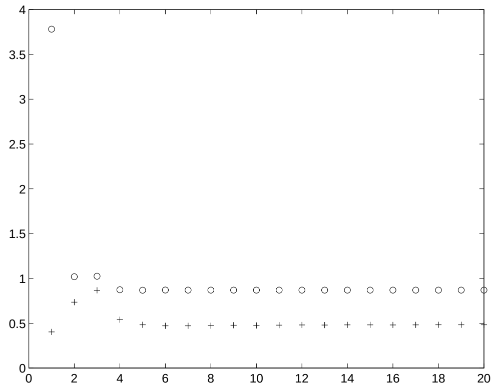

Figure 5: SQR: $A - BF_i$ (o) and $V_i'(A - BF_i)V_i$ (+) , Example 1

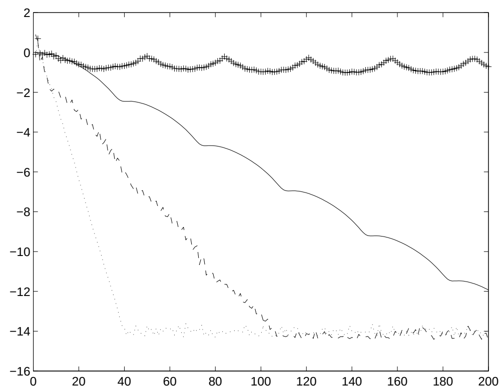

Figure 6: SQR: Feedback (- -), V_i (.), SSI (+) and $\prod(A - BF_i)B$ (–) , Example 1

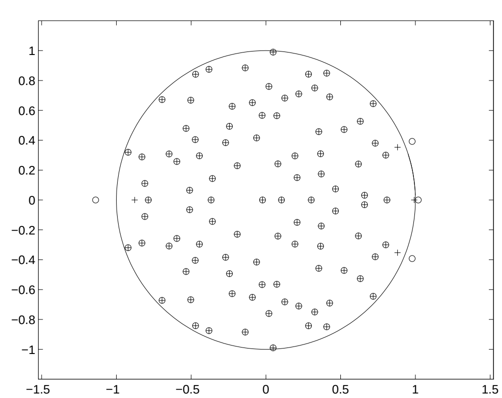

Figure 7: SQR: Spectrum of A (o) and $A - BF_{300}$ (+), Example 2

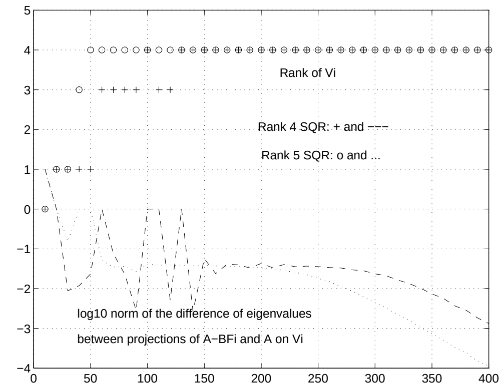

Figure 8: SQR: $\log_{10}(\Delta V_i)$, Example 2

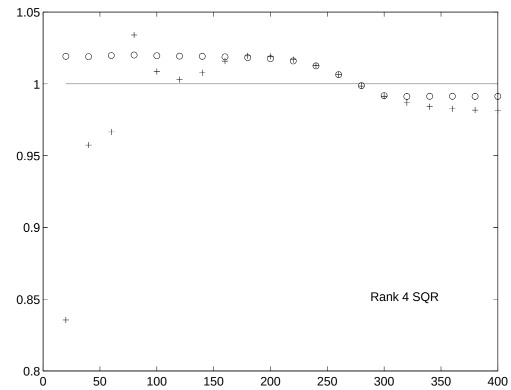

Figure 9: SQR: $A - BF_i$ (o) and $V_i'(A - BF_i)V_i$ (+), Example 2

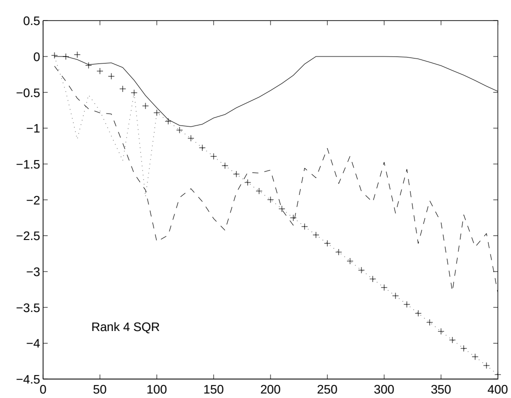

Figure 10: SQR: Feedback (- -), SSI (.) and $\prod(A - BF_i)B$ (–), Example 2

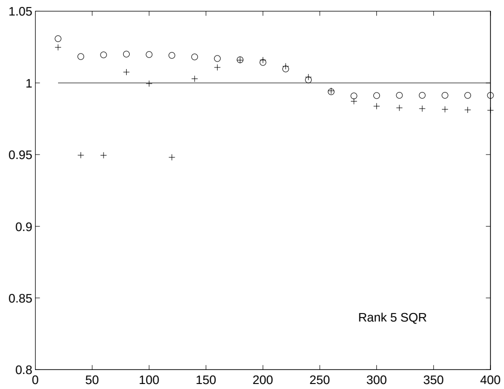

Figure 11: SQR: $A - BF_i$ (o) and $V_i'(A - BF_i)V_i$ (+), Example 2

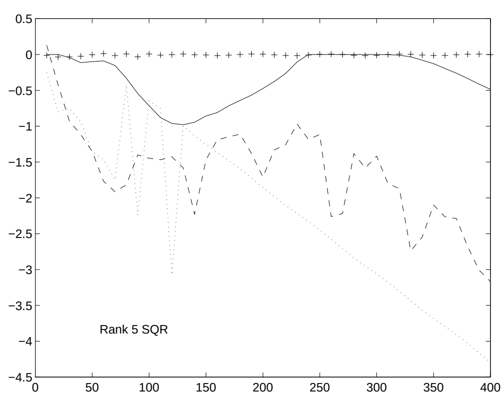

Figure 12: SQR: Feedback(- -), V_i (.), SSI (+) and $\prod(A - BF_i)B$ (–) , Example 2

a few steps, while the residuals of the approximated unstable subspace from SSI have not reached significant accuracy. We can relax the condition that all eigenvalues of A are well-separated from the unit circle to that all unstable eigenvalues of A are well-separated from all stable eigenvalues of A. If the unstable eigenvalues of A are well-separated from the unit circle, fast stabilization with SQR is expected and feedback convergence with SQR depends on the choice of the rank of $P_0^{1/2}$, with the worst case when some stable eigenvalues of A are very close to the unit circle and we choose an incorrect rank of $P_0^{1/2}$ (larger than the number of unstable eigenvalues of A). In this case, we can monitor the eigenvalue convergence of $V_i'(A - BF_i)V_i$ to catch the stability of $A - BF_i$ or modify the rank of $P_i^{1/2}$ during the iteration. If some unstable eigenvalues of A are very close to the unit circle and stable eigenvalues of A are well-separated from the unit circle, some scaling on A can help to accelerate both stabilization and feedback convergence.

From Example 2, we see the limitations of SQR when applied to a system which is less well-conditioned and A has many stable and unstable eigenvalues which are close to the unit circle. For such problems both the subspace iteration associated with SQR and feedback convergence are expected to converge slowly (see [6] for a more detailed discussion).

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

This scaling technique works very well if there exists a $r < 1$ such that the number of eigenvalues of A with absolute value not less than r 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 r). For such a special case, the stabilization and feedback convergence with general SQR are very slow, but scaling A, without changing the rank of $P_0^{1/2}$, will accelerate 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. These stable eigenvalues must be moved along with the unstable eigenvalues in order to efficiently stabilize the system. This increase in the number of eigenvalues moved by $A - BF$ with scaling r influences the choice of rank of $P_0^{1/2}$ and suggests we may need to increase the rank of $P_0^{1/2}$ so that the rank is consistent to number of unstable eigenvalues of A/r. If this can be done then efficient stabilization is possible. The stable portion of the original spectrum, however, is modified in contrast to SQR or SSI on the original system. These diffi-

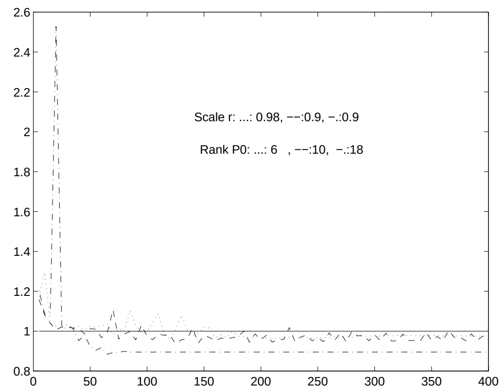

Figure 13: SQR with scaling A: $A - BF_i$, Example 3

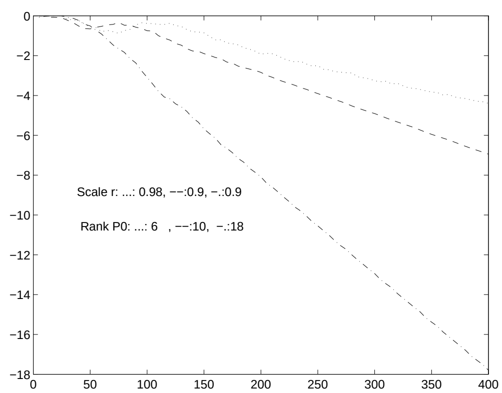

Figure 14: SQR with scaling A: $\prod(A - BF_i)B$,Example 3

culties can make the application of scaling to SQR difficult for the general problem.

Example 3 *A, B are the same as in Example 2. r is chosen as 0.98, 0.9 and 0.9, the rank of $P_0^{1/2}$ is chosen 6, 10, 18. Figure 13 shows the spectral radius of $A - BF_i$ and Figure 14 shows the stabilization criterion $\log_{10} \|(\prod_{k=1}^{i}(A - BF_k))B\|_2$. SQR with scaling A takes much less SQR iteration steps to stabilize. The vibration in Figure 13 is from the bad rank choice (10) of $P_0^{1/2}$ which is inconsistent with $r = 0.9$. Note the rank of $P_0^{1/2}$ is significantly larger than the number of unstable eigenvalues of A, which is the major disadvantage of SQR and SSI.*

6 Comparison of SQR and SSI

For SQR, we focus on feedback convergence and the ΔV_i stabilization criterion to check convergence. For SSI, we focus on subspace residuals. So, SQR and SSI performance can often be difficult to compare in general. Some clear advantages are present in SQR however.

1. The feedback F_i can be computed from $(R_i^{\epsilon})^{1/2}$ and $\tilde{K}_i$ cheaply compared to SSI.
2. For stabilization, the feedback convergence check of $\|F_i - F_{i-1}\|$ is therefore very cheap for SQR. Computing subspace residual check costs the same for SQR and SSI.
3. The use of ΔV_i can detect the stability of $A - BF_i$ much earlier than residual convergence check of SSI or feedback convergence check of SQR since it is not known how accurate the subspace residuals should be in order to stop SSI iterations with a high probability of stabilization.
4. SQR can produce the same subspace sequence as SSI with only the additional cost of low order QR decomposition. As a result, if feedback convergence is too slow, we can always check the associated subspace V_i for adjustment and projection, i.e., SQR has the ability to shift into SSI mode for problems where SQR has inferior convergence.
5. For well-conditioned stabilization problems where the eigenvalues of A are well-separated from the unit circle, the rank choice in SQR is not as critical as it is in SSI. For stabilization purposes, both SQR and SSI need a rank no less than the number of unstable eigenvalues. If an overestimate of the rank is chosen for $P_0^{1/2}$, SQR works fine. SSI, however, may need special additions such as SRR to maintain acceptable convergence.
6. As the number of inputs increases, the stabilization problem for (A, B) improves in conditioning and SQR will need fewer iterations to achieve stabilization. SSI ignores B and the link between A and B and therefore cannot benefit from the increased number of inputs.

We have seen that SQR is superior to SSI-based methods for many problems. As with projection methods, SQR only moves unstable eigenvalues of A (in this case to their reciprocals). The ranks of the SQR and SSI algorithms are determined by the number of unstable eigenvalues of A and the distribution of eigenvalues of A. For a well-conditioned system with only few unstable eigenvalues that are well-separated from the unit circle, SQR can stabilize the system quickly. If there exists a circle with radius $r < 1$ such that the number of eigenvalues outside the circle is very small, scaling A can accelerate stabilization process significantly at the cost of possibly increasing the rank of the algorithm and thereby requiring a more careful rank choice heuristic. Because SQR can generate the same subspace sequence as SSI, in case the SSI residual convergence is much faster than SQR feedback convergence, we can always implement a low order QR decomposition to extract an approximate unstable invariant subspace from SQR and then continue with a projection method.

When using SQR, with or without scaling, we have noticed that the system is stabilized much earlier than any traditional stopping criterion is reached. The ΔV criterion has improved this situation but, further research on efficient stopping criteria is needed.

With SQR we have successfully created an algorithm that integrates the two phases of projection methods and improves the convergence rate by reducing the sensitivity to the choice of rank of the algorithm. For a less well-conditioned system which cannot separate unstable eigenvalues from stable eigenvalues, and the number of eigenvalues close to the unit circle is significant, or for a system that has many unstable eigenvalues, the rank of both SQR and SSI is too high, or the number of iterations required for stabilization is too large. The large rank of SQR and SSI implies a lack of scalability in the algorithm even though it is known that there are well-conditioned stabilization problems with large numbers of unstable eigenvalues.

7 The TSQR Stabilization Algorithm

We have also developed a second SQR algorithm that can be useful in some circumstances where SQR has difficulties and that can be used as the basis for a general DRE solver. It is based on an alternative form of the DRDE

$$P_{i+1} = (A - BF_i)'P_i(A - BF_i) + F_i'RF_i + Q \qquad (5)$$

where $P_i = (R + B'P_iB)^{-1}B'P_iA$ and derived by decomposing every term in (5) into its square root form. First form the matrix

$$\left[(A - BF_i)'P_i^{1/2} \;\; F_i'R^{1/2} \;\; Q^{1/2}\right]$$

and use the economical SVD to decompose it into the form $U_iS_iV_i$ where the diagonal elements of S_i are ordered with larger ones on top. The matrix $P_{i+1}^{1/2}$ is formed by the first several columns of U_i and corresponding submatrix of S_i (hence this algorithm is referred to the truncated square root algorithm – TSQR). With $Q = 0$, this truncated algorithm is equivalent to SQR of the same rank. However, by using nonzero Q, the spectral radius of $A - BF$ can be influenced and thereby the convergence of the iteration. Care must be taken with TSQR when Q is nonzero to avoid certain numerical problems. Unless the rank of $P_i^{1/2}$ is large enough, TSQR can have trouble with convergence and stability. These difficulties can be solved by limiting the iteration to the unstable left invariant subspace of A, which can be achieved by updating Q with an orthogonal iteration on each iteration of (5). As a result, in practice, the rank of the basis matrix propagated by TSQR is usually slightly larger than the number of unstable eigenvalues of A and that used by SQR.

The resulting SQR and TSQR methods essentially propagate a low-rank approximation to a basis of the invariant subspace and information needed to cheaply compute the stabilizing feedback. When parameters are carefully chosen the convergence is less sensitive than that of SSI/SRR and performance is as least as good and often significantly better than Saad's method.

8 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.

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 another well-known recurrence called the square root Chandrasekhar [2] algorithm (noted as CSQR). It is based on the DRDE with $P_0 = 0$.

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

$$\begin{pmatrix} (R_{i-1}^\epsilon)^{1/2} & B'L_{i-1} \\ \tilde{K}_{i-1} & A'L_{i-1} \end{pmatrix} U_i = \begin{pmatrix} (R_i^\epsilon)^{1/2} & 0 \\ \tilde{K}_i & L_i \end{pmatrix} \qquad (6)$$

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 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. In [6], techniques that improve its performance significantly are discussed and these will be presented in a forthcoming paper.

For rank l SSI (measured by rank of initial subspace), SQR and TSQR (measured by rank of $P_0^{1/2}$), CSQR (measured by rank of L_i), the computation costs of one step iteration are roughly close, $(n + p)nl + (p + l)p^2 + n(p + l)^2$ flops for dense (A, B) and $O(\rho n)l + (p + l)p^2 + n(p + l)^2$ flops for sparse (A, B) with an average nonzero count per row of ρ, while a QR decomposition on an $n \times n$ matrix needs $O(n^3)$ computations. Note the rank of subspace iteration, SQR and TSQR is at least the number of unstable eigenvalues of A, and the rank of CSQR is at most the maximum multiplicity of unstable eigenvalues of A and usually can be chosen as 1 (or 1 can be used with deflation), so CSQR will be the best choice if no knowledge about the number of unstable eigenvalues exists or this number is too large.

9 Conclusion

In general, the results here and in [6] show these methods have tremendous promise to efficiently stabilize large scale systems. However, much remains to be done. The SQR method, for example, can be viewed as an efficient integration of feedback computation and an eigensolver midway between the power method and subspace iteration. There are indications that the iterations to solve the DRDE, such as

SQR, CSQR and an inexact Newton variant we have also developed are flexible enough to subsume other more sophisticated eigensolvers. This is currently under investigation. Unquestionably, the SQR, TSQR, CSQR family of methods is one of the most promising and efficient stabilization approaches available for large scale systems.

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

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

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

[5] Chungyang He and Volker Merhmann: *Stabilization of large linear systems*

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