Efficient 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

Abstract

In this paper we discuss the stabilization of large scale linear time invariant dynamical systems via feedback. An overview of efficient schemes based on the Discrete Riccati Difference Equation are presented. In particular, results are given for a Newton-like approach to the problem.

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 corresponding system 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 complexity of standard algorithms for these approaches is $O(n^3$ and hence 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 Saad's projection method [1], in which stabilization or eigenvalue assignment is only imposed on a small invariant subspace. 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. 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 $F = F_u V'$ for the original system (A, B). Such an approach is often effective, but it can have convergence difficulties and the need for a basis of a left invariant subspace V' of A (with presumably small dimension), that contains the left unstable invariant subspace of A can cause excess space requirements for very large systems or for systems where the number of unstable eigenvalues is not small.

In this paper, we summarize some efficient alternatives that address the convergence difficulties and motivate an 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 [2].

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)

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

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

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) exists 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 [2] 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 [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 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 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).

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 the 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 detect 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 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 CSQR Stabilization Algorithm

SQR addresses the first two difficulties with Saad's method (improved convergence and stability detection, and unifies the two stages). However, since l, the rank of the 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 [4] 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

$$\begin{pmatrix} (R_{i-1}^{\epsilon})^{1/2} & B'L_{i-1} \\ \tilde{K}_{i-1} & A'L_{i-1} \end{pmatrix} U_i = \begin{pmatrix} (R_i^{\epsilon})^{1/2} & 0 \\ \tilde{K}_i & L_i \end{pmatrix} \qquad (5)$$

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 major difficulties with Saad's method and in the form above is practical for many problems. In [2], techniques that improve its performance significantly are discussed and these will be presented in a forthcoming paper.

6 The TSQR Stabilization Algorithm

The low-rank algorithms SQR and CSQR that stabilize a large scale (sparse) discrete time system are two different implementations of the DRDE with different parameters and initial conditions. The DRDE is a *fixed point* iteration method to solve the DRE and converges linearly with the rate $\rho^2(A - BF_\infty)$. Newton's method, in theory, has quadratic convergence when used to solve non-linear equations. In the remainder of this paper, we first summarize Newton's iteration for the DRE then describe a new square root algorithm, TSQR, as an implementation of an approximate Newton's iteration, of which the DRDE is a special case. Finally, some results are given indicating the promise and problems with the TSQR method.

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]):

$$X_i - A'_{i-1} X_i A_{i-1} = Q + F'_{i-1} R F_{i-1}, i = 1, \cdots, \quad (6)$$

where

$$F_i = (R + B' X_i B)^{-1} B' X_i A, \ A_i = A - BF_i \quad (7)$$

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

Theorem 1 *Let (A, B) be stabilizable. Assume that there is a symmetric solution X of the inequality $\mathcal{R}(X) \geq 0$ for which $R + B'XB > 0$. For any F_0 such that $A_0 = A - BF_0$ is stable and X_0 which satisfies*

$$X_0 - A'_0 X_0 A_0 = Q + F'_0 R F_0$$

the iteration (6) determines a decreasing sequence of symmetric matrices X_i such that $A - BF_i$ is stable and X_i converges to X_+, a maximal symmetric solution of DRE. (Maximal means that for any symmetric solution X of DRE, $X_+ - X$ is nonnegative.) Moreover, $A - B(R + B'X_+B)^{-1}B'X_+A$ is stable and there is a constant $c > 0$ such that $\|X_{i+1} - X_+\| \leq c\|X_i - X_+\|^2$ for $i = 0, 1, \cdots$ and any matrix norm $\|\cdot\|$.

Although Newton's iteration converges quadratically, the implementation is not easy. 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; and the cost to solve the Lyapunov equation is almost the same as solving the DRDE. An approximate implementation of Newton's method must be developed to address these problems. The starting point for this algorithm is:

Algorithm 1 Approximate Newton Iteration for DRE: *Specify a small integer K (usually around 5). Starting from $X_0^K = X_0$, a positive semidefinite random matrix, for $i = 1, 2, \cdots$, (outer loop) and $j = 1, 2, \cdots, K$ (inner loop),*

$$X_i^j = A'_{i-1} X_i^{j-1} A_{i-1} + Q + F'_{i-1} R F_{i-1} \quad (8)$$

where $X_i^0 = X_{i-1}^K$, $F_{i-1} = (R + B'X_{i-1}^K B)^{-1} B' X_{i-1}^K A$, $A_{i-1} = A - BF_{i-1}$.

Algorithm 1 does not require a stabilizing initial guess since we only use finite steps of iteration to approximate each Lyapunov equation. We need only randomly choose a positive semidefinite X_0. 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 Algorithm 1 stabilizes and converges much faster than the DRDE-based CSQR algorithm, because we can freely choose the initial guess while Chandrasekhar algorithm restricts its initial guess to zero. If K is large enough, the convergence is quadratic. (measured by number of outer loops used).

Direct implementation of Algorithm 1 can be as unstable as the direct implementation of the DRDE. Note that all the terms in both sides of Equation (8) should be positive semidefinite. So, we can borrow ideas from CSQR and SQR to keep X_i^j positive semidefinite. After modification Equation (8) becomes

$$\left[A'_{i-1}(X_i^{j-1})^{1/2} \ \ Q^{1/2} \ \ F'_{i-1}R^{1/2}\right] U = \left[(X_i^j)^{1/2} \ \ 0\right] \quad (9)$$

where U is orthogonal.

For large scale systems, especially large sparse systems, a full dimension implementation of Equation (9) with an $n \times n$ X_i^j is computationally 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. Usually, 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 key to using a square root and truncation algorithm for stabilization is the choice of l. It cannot be too small, at least it should be larger than the number of unstable eigenvalues. Even so, since an eigenvalue being unstable does not necessarily mean that it will have larger components in X_i^j than stable ones, we have to be very careful in choosing l. Another important issue is the convergence of the truncation algorithm, which needs further research even for the solution of the DRDE, i.e., independent of stabilization. For stabilization purposes, if l is not smaller than the number of unstable eigenvalues and the spectral radius of the corresponding closed loop matrix is not too close to 1, we can use $(A - BF_i) \cdots (A - BF_1)B$ to measure the stabilization. One exception is when $Q = 0$ where the rank of a stabilizing solution of DRE is determined by the number of unstable eigenvalues, so we can use the stabilization/convergence measures we used for SQR.

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

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

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

      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.*

7 Experiments

In this section we consider two examples from [2]. First, consider an example where $Q = 0$ (so truncation will not cause trouble with convergence if the rank of $(X_i^j)^{1/2}$ is at least the number of unstable eigenvalues).

Example 1 *The matrix A is a 100×100 matrix generated by RANDN of MATLAB and is scaled to have a spectrum with many eigenvalues (both stable and unstable) near the unit circle. The matrix B is a 100×1 matrix generated by RANDN of MATLAB. Such a problem is expected to cause difficulties for techniques such as SQR. $(X_0)^{1/2}$ is generated as a 100×4 random matrix by RAND of MATLAB. Figure 1 plots the spectral radius of the corresponding closed-loop matrix vs total number of steps of TSQR (inner loop $\times$ outer loop). Figure 2 shows the feedback convergence (here we use $F_i - F_\infty$ instead feedback increment). Note that $K = 1$ is the DRDE. Measured by the total number of iterations, in the case where $K = 5$ or $K = 10$, both the spectral radius of corresponding closed-loop and feedback converges much faster than the DRDE. For the corresponding closed-loop spectral radius to reach convergence, the DRDE uses at least 300 steps while TSQR with $K = 5$ or $K = 10$ uses less than 100. Furthermore, from Figure 2, we can see that TSQR with $K = 5$ or $K = 10$ starts to converge at the beginning of the iteration while the DRDE spends some iterations adjusting. The reason is, some unstable eigenvalues of A are very close to 1, and TSQR with larger K gives these eigenvalues more opportunities to increase their components in $(X_i^j)^{1/2}$ and F_i. When applied a system for which all the eigenvalues are well-separated from unit circle, TSQR with larger K uses more steps that the DRDE to converge to the same tolerance.*

With nonzero Q, TSQR is very sensitive to the choice of l. In Example 1, if a nonzero $Q^{1/2}$ is used keeping all other parameters fixed, the case with $K = 1$ (DRDE) needs a very large l to reach a stabilizing feedback. Even when we choose $K = 5$ or $K = 10$, we still need to choose K larger than the number of unstable eigenvalues and need large number of iterations to reach a stabilizing feedback. The reason is, if Q is not orthogonal to a right stable eigenvector, we can not assume the component of this eigenvector in P or X_+ is less significant than unstable eigenvectors. TSQR sometimes performs poorly for stabilization since the components of some unstable eigenvectors in (X_i^j) and F_i are truncated. However, if and only if $Q^{1/2}$ only contains unstable components of A, that is, $v'Q^{1/2} = 0$ for any right stable eigenvector of A, the solution of P or X_+ of DRE should not contain any components of stable eigenvectors of A. So, truncation should be able to keep the unstable components instead of the stable components.

TSQR could be modified to improve this situation by inserting SSI on $Q^{1/2}$ along with each $(X_i^j)^{1/2}$ (inner) iteration such that $(X_i^j)^{1/2}$ will only focus on the unstable components of A. We can also only insert the SSI on Q into the outer loop along with the update of F_i. For $K = 1$, this method is equivalent to solving a DRDE $P_i = A'(P_{i-1} - P_{i-1}B(R + B'P_{i-1}B)^{-1}B'P_{i-1})A + Q_{i-1}$ where $Q_{i-1}^{1/2}$ is updated via SSI on $Q_0^{1/2}$ with A'. Since SSI will converge to eigenvectors with the largest eigenvalues, for best performance, we should choose the rank of $Q_0^{1/2}$ the same as the number of unstable eigenvalues of A so that all unstable eigenvectors of A will have significant components in Q_i. On the other hand, it is better to choose l larger than the number of unstable eigenvalues for higher flexibility in truncation. Example 2 shows that the performance of both stabilization and feedback convergence are enhanced significantly with this type of modification on Q if the rank of $(X_i^j)^{1/2}$ is not less than the number of unstable eigenvalues of A.

Example 2 *The system (A, B) and $(X_0)^{1/2}$ is the same as in Example 1. $Q_0^{1/2}$ is a randomly chosen 100×4 matrix and l is chosen as 6. $Q^{1/2}$ is updated by SSI as discussed. Compared to Example 1, Figure 3 shows that with SSI on Q, the total number of iterations to reach a stabilizing feedback is cut almost by half for $K > 1$, while feedback convergence rate (Figure 4) does not change significantly.*

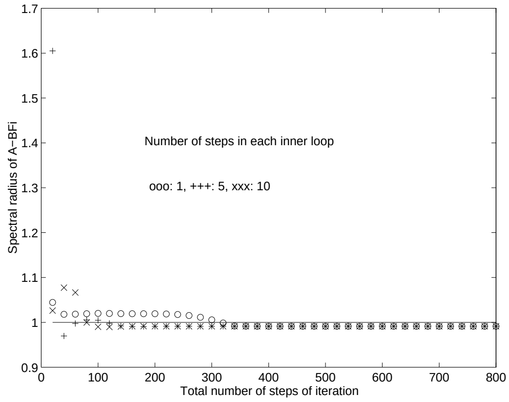

Figure 1: TSQR($Q = 0$) vs. SQR: Spectral radii of $A - BF_i$, Example 1

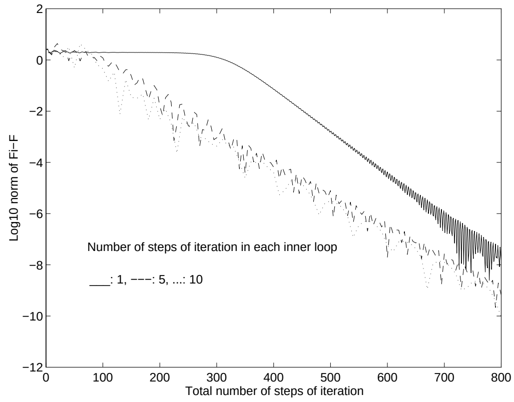

Figure 2: TSQR($Q = 0$) vs. SQR: Feedback convergence, Example 1

8 Summary

We have summarized three alternatives to Saad's method for stabilization of large sparse dynamical systems. We have concentrated on describing a low-rank stabilization algorithm, TSQR, based on Newton's method to solve the DRE. Details of the analysis and performance of SQR and CSQR can be found in [2].

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 CSQR algorithms. TSQR also unifies our algorithms and others in the literature such as [7].

Many open questions remain. The first is the convergence properties of the approximate Newton's iteration (Algorithm 1) for an arbitrary initial condition. The second is the convergence properties of TSQR for both stabilization and solution of the DRDE. The current form of TSQR does not attempt to exploit techniques that were used to transform SQR into CSQR, i.e., computing the difference rather than the square root. Considering extensions of TSQR along those lines is promising. The approximation of the solution to the Lyapunov equation required on each step must also be considered further in terms of efficiency and its overall role in the convergence of the iteration. Finally, the approximate Newton's iteration and TSQR may be very useful when applied to solving the continuous-time Riccati Equation for various purposes. Their adaptation to this case should be considered. Further research on these issues will benefit not only stabilization in optimal control, but also optimal filtering and data assimilation problems where solving a large scale DRDE might be impractical.

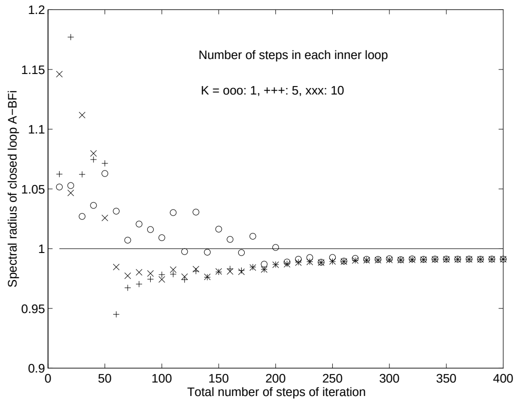

Figure 3: TSQR: Spectral radii of $A - BF_i$, Example 2

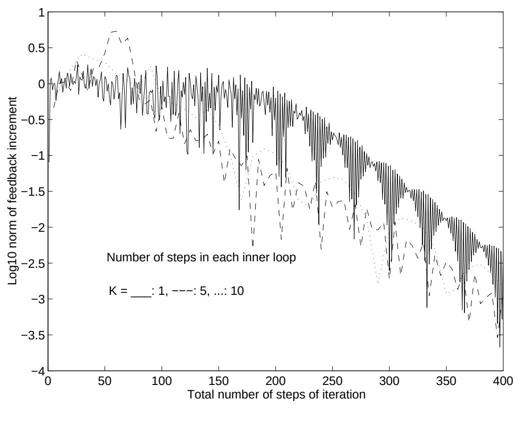

Figure 4: TSQR: Feedback convergence, Example 2

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.