

# A new block method for computing the Hamiltonian Schur form <sup>\*</sup>

V. Mehrmann<sup>‡</sup>, C. Schröder<sup>‡</sup> and D. S. Watkins<sup>§</sup>

September 8, 2008

**Dedicated to Henk van der Vorst on the occasion of his 65th birthday**

## Abstract

A generalization of the method of Chu, Liu and Mehrmann [7] for the computation of the Hamiltonian real Schur form is presented. The new method avoids some of the difficulties that may arise when a Hamiltonian matrix has tightly clustered groups of eigenvalues. A detailed analysis of the method is presented and several numerical examples demonstrate the superior behavior of the method.

**Key words.** linear-quadratic control,  $H_\infty$  control, Hamiltonian matrix, Hamiltonian real Schur form, symplectic matrix, eigenvalue cluster

**AMS subject classification.** 65F15, 15A21, 93B40.

## 1 Introduction

The need to compute the Hamiltonian Schur form of a Hamiltonian matrix arises in several applications in control theory, in particular in the linear-quadratic control problem [12, 17] and the  $H_\infty$  control problem [23]. The problem of developing an  $O(n^3)$  backward-stable algorithm that preserves the Hamiltonian structure, which was posed by Paige and Van Loan [18] in 1981, proved difficult to solve however, so much so that it came to be known as *Van Loan's curse*. In 2004 Chu, Liu, and Mehrmann [7], see also [21], a new method was proposed, which we will call *CLM*, that seems to lift that curse. In [7] it is shown that CLM performs well on a wide variety of benchmark problems from control theory [4]. The method performs well, in particular, for Hamiltonian matrices with eigenvalues near or on the imaginary axis for which other methods that

---

<sup>\*</sup>Partially supported by *Deutsche Forschungsgemeinschaft*, through the DFG Research Center MATHEON *Mathematics for Key Technologies* in Berlin.

<sup>‡</sup>Institut für Mathematik, TU Berlin, Str. des 17. Juni 136, D-10623 Berlin, FRG. {mehrmann,schroed}@math.tu-berlin.de.

<sup>§</sup>Department of Mathematics, Washington State University, Pullman, WA 99164-3113, USA. watkins@math.wsu.edu.

do not preserve the structure have difficulties. However, we have found subsequently that it is possible to contrive Hamiltonian matrices (not from actual control problems) that cause CLM to perform badly. These are matrices that have large tightly-packed clusters of eigenvalues away from the imaginary axis. Interestingly, these represent “easy” problems from the standpoint of control theory. In a nutshell, one can say that CLM performs badly on these problems, because it implicitly performs a swapping of eigenvalues [21] and it is known that the problem of swapping eigenvalues that are tightly packed may lead to large errors.

The existence of such difficult problems for the current version of CLM, demonstrates the need to develop a more general and comprehensive strategy that is able to deal with this class of extreme problems and perhaps other classes as well that could arise in applications in the future. In this paper we take a step in that direction by introducing a generalization of CLM that processes the eigenvalues in blocks. Eigenvalues belonging to a tight cluster are placed in the same block, and no attempt is made to distinguish them from one another. This enables the accurate solution of problems containing tight clusters of eigenvalues.

A second advantage of the block algorithm is that, if the blocks are large, the bulk of the arithmetic work of the algorithm consists of matrix multiplications in which the dimensions of the matrices are large enough to make efficient use of cache memory and obtain thereby high performance [11].

The paper is organized as follows. In Section 2 we introduce the basic notation and discuss some preliminary results. In Section 3 we present the three slightly different computational problems associated with Hamiltonian matrices that we address. The new algorithm is described in Section 4 and in Section 5 we analyze why the algorithm works, with details in Section 6. Numerical results are presented in Section 7.

## 2 Definitions and Preliminary Results

Throughout this paper we restrict our attention to matrices with real entries. Define  $J \in \mathbb{R}^{2n \times 2n}$  by

$$J_n = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}. \quad (1)$$

We leave off the subscript  $n$  if it is clear from the context.

A matrix  $H \in \mathbb{R}^{2n \times 2n}$  is called *Hamiltonian* if  $(JH)^T = JH$ ,  $K \in \mathbb{R}^{2n \times 2n}$  is called *skew Hamiltonian* if  $(JK)^T = -JK$ , and  $S \in \mathbb{R}^{2n \times 2n}$  is called *symplectic* if  $S^T J S = J$ .

A matrix  $H \in \mathbb{R}^{2n \times 2n}$  is Hamiltonian if and only if

$$H = \begin{bmatrix} A & G \\ C & -A^T \end{bmatrix}, \quad \text{where } G^T = G \text{ and } C^T = C.$$

A subspace  $\mathcal{S} \subseteq \mathbb{R}^{2n}$  is *isotropic* if  $x^T J y = 0$  for all  $x, y \in \mathcal{S}$ . Let for  $S_1 \in \mathbb{R}^{2n \times k}$ ,  $\mathcal{S} = \mathcal{R}(S_1)$  denote the range of  $S_1$ , then  $\mathcal{S}$  is isotropic if and only

if  $S_1^T JS_1 = 0$ . If  $S$  is symplectic and  $S = \begin{bmatrix} S_1 & S_2 \end{bmatrix}$ , where  $S_1, S_2 \in \mathbb{R}^{2n \times n}$ , then  $S_1^T JS_1 = 0$  and  $S_2^T JS_2 = 0$ . Thus  $\mathcal{R}(S_1)$  and  $\mathcal{R}(S_2)$  are isotropic. If  $\mathcal{S}$  is isotropic and  $Q$  is symplectic, then  $QS$  is isotropic.

If  $H$  is Hamiltonian and  $x$  is a (right) eigenvector of  $H$  associated with eigenvalue  $\lambda$ , then  $(Jx)^T$  is a left eigenvector of  $H$  associated with eigenvalue  $-\lambda$ . Thus the eigenvalues of a Hamiltonian matrix occur in pairs  $(\lambda, -\lambda)$ . If there are no eigenvalues on the imaginary axis, then there must be exactly  $n$  in the left half plane and  $n$  in the right half plane.

The following result is well-known, see e.g. [8].

**Theorem 1** *Let  $H \in \mathbb{R}^{2n \times 2n}$  be Hamiltonian, let  $\mathcal{S} \subseteq \mathbb{R}^{2n}$  be a subspace that is invariant under  $H$ , and let  $\Lambda$  be the set of eigenvalues of  $H$  associated with the subspace  $\mathcal{S}$ . If  $\Lambda$  does not contain any pairs of eigenvalues of the form  $(\lambda, -\lambda)$ , then  $\mathcal{S}$  is isotropic.*

For example, if in Theorem 1 all elements of  $\Lambda$  have negative real part, then  $\mathcal{S}$  is isotropic.

The algorithm discussed in this paper employs similarity transformations. If  $H$  is Hamiltonian (skew-Hamiltonian) and  $S$  is symplectic, then  $S^{-1}HS$  is also Hamiltonian (resp. skew Hamiltonian). Thus, Hamiltonian structure is preserved by symplectic similarity transformations. In the interest of numerical stability we prefer to work with orthogonal similarity transformations. Thus we will allow only similarity transformations by matrices that are both orthogonal and symplectic. These are matrices  $Q \in \mathbb{R}^{2n \times 2n}$  that satisfy both  $Q^T Q = I$  and  $Q^T J Q = J$ .  $Q$  is orthogonal and symplectic if and only if

$$Q = \begin{bmatrix} Q_1 & -Q_2 \\ Q_2 & Q_1 \end{bmatrix},$$

where  $Q_1^T Q_1 + Q_2^T Q_2 = I$  and  $Q_1^T Q_2 - Q_2^T Q_1 = 0$ , in other words, the columns of  $\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$  are orthonormal and span an isotropic subspace of  $\mathbb{R}^{2n}$ . As a special case, if  $Q$  has the form

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_1 \end{bmatrix},$$

where  $Q_1^T Q_1 = I$ , then  $Q$  is orthogonal and symplectic.

A Hamiltonian matrix  $H$  is in *Hamiltonian real Schur form* if

$$H = \begin{bmatrix} T & G \\ 0 & -T^T \end{bmatrix}, \quad (2)$$

where  $T$  is *quasitriangular*, that is,  $T$  is block upper triangular with  $1 \times 1$  and  $2 \times 2$  blocks on the main diagonal. Each  $1 \times 1$  block is a real eigenvalue of  $H$ , while each  $2 \times 2$  block houses a complex conjugate pair of eigenvalues.

Necessary and sufficient conditions for the existence of a Hamiltonian real Schur form in terms of the Jordan structure of  $H$  are well known, see [1, 8, 15, 19].

**Theorem 2** *Let  $H \in \mathbb{R}^{2n \times 2n}$  be a Hamiltonian matrix, let  $i\alpha_1, \dots, i\alpha_\nu$  be its pairwise distinct purely imaginary eigenvalues, and let the columns of  $U_k \in \mathbb{C}^{2n \times m_k}$ ,  $k = 1, \dots, \nu$ , span the associated invariant subspaces of dimension  $m_k$ . Then the following are equivalent:*

- (i) *There exists an orthogonal symplectic matrix  $U \in \mathbb{R}^{2n \times 2n}$  such that  $U^T H U$  is in Hamiltonian real Schur form.*
- (ii)  *$m_k$  is even and  $U_k^H J U_k$  is congruent to  $J_{m_k/2}$  for all  $k = 1, \dots, \nu$ .*

*In particular, there exists a Hamiltonian real Schur form if  $H$  has no purely imaginary eigenvalues.*

Typically in applications one is interested in the case where the eigenvalues of the matrix  $T$  in the Hamiltonian real Schur form are all in the (closed) left half plane. However, the ordering of the eigenvalues on the block-diagonal can also be arranged so that  $T$  has a mixture of left and right half plane eigenvalues within certain obvious limits: If  $\lambda$  is an eigenvalue of  $T$ , then  $-\lambda$  is an eigenvalue of  $-T^T$ . Therefore, if (for example)  $\lambda$  is a simple eigenvalue of  $H$ , then  $T$  cannot have both  $\lambda$  and  $-\lambda$  as eigenvalues.

From these preliminary results it is clear that several difficulties may arise if we want to compute the Hamiltonian real Schur form with a numerical method. First of all it will become necessary to distinguish between purely imaginary eigenvalues and those eigenvalues with nonzero real part.

For this it is essential to use structure preserving transformations, since only these can guarantee that well-conditioned purely imaginary eigenvalues stay purely imaginary, see [16, 19] for a detailed perturbation analysis. On the other hand, the existence of a Hamiltonian real Schur form (as described in Theorem 2) implies that purely imaginary eigenvalues have to occur in either even sized Jordan blocks or an even number of odd sized Jordan blocks. This typically means that these eigenvalues are ill-conditioned, i.e. their computation presents a challenge for numerical methods, because under small (structured) perturbations (in finite precision arithmetic) this Jordan block will split and lead almost surely to eigenvalues with small nonzero real parts. Thus, we cannot distinguish these eigenvalues from those near to the imaginary axis. On the other hand, well separated simple eigenvalues on the imaginary axis, or those multiple purely imaginary eigenvalues where the condition (ii) in Theorem 2 does not hold (see [16]) are robust under small perturbations and thus it is possible to decide whether they are on the axis in finite precision arithmetic.

If we are interested in computing an existing Hamiltonian real Schur form, however, then those Jordan blocks associated with purely imaginary eigenvalues that are bound to split into pairs of eigenvalues with nonzero real part may not cause a problem, since round-off errors will almost surely lead to the existence of a nearby Hamiltonian real Schur form.

Nevertheless it will make sense to cluster these eigenvalues together with other nearby eigenvalues.

If there are simple eigenvalues on the imaginary axis or multiple eigenvalues, where the condition (ii) in Theorem 2 does not hold, then we cannot compute

a Hamiltonian real Schur form and thus in this case it makes sense to separate these parts of the spectrum from the remaining spectrum.

In the next section we summarize this discussion by describing the computational tasks that we address.

### 3 Problems that will be addressed

Starting from a Hamiltonian matrix  $H$ , we compute a partial Hamiltonian real Schur form

$$H_{\text{final}} = Q^T H Q = \left[ \begin{array}{cc|cc} T_{11} & T_{12} & G_{11} & G_{12} \\ 0 & T_{22} & G_{21} & G_{22} \\ \hline 0 & 0 & -T_{11}^T & 0 \\ 0 & C_{22} & -T_{12}^T & -T_{22}^T \end{array} \right], \quad (3)$$

where  $T_{11}$  is *block triangular* and has eigenvalues that are away from the imaginary axis, (typically the block is chosen so that the eigenvalues are in the open left half plane), and the Hamiltonian block

$$\left[ \begin{array}{cc} T_{22} & G_{22} \\ C_{22} & -T_{22}^T \end{array} \right] \quad (4)$$

contains elements that we are unable to reduce further. It may have only purely imaginary eigenvalues, or it may have eigenvalues that are near the imaginary axis that cannot be distinguished from purely imaginary eigenvalues, or it may have some of each.

Based on this form we can solve three slightly different computational problems. The first is to transform a Hamiltonian matrix to the form (3), where we might not care whether the eigenvalues of  $T_{11}$  all lie in the left half plane. If this is the case, then no ordering is necessary, but we note that if the eigenvalues of  $T_{11}$  appear in some order that is different from what we want, then it is possible to change the order.

If  $H$  has no eigenvalues on or very near the imaginary axis, then as our second problem, we consider computing the Hamiltonian real Schur form (2), where the eigenvalues of  $T$  all lie in the left half plane. Then if we let  $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$  with  $Q_1 \in \mathbb{R}^{2n \times n}$ , then  $HQ_1 = Q_1T$ . This shows that  $\mathcal{R}(Q_1)$  is the invariant subspace under  $H$  corresponding to the eigenvalues that lie in the left half plane. This is called the *stable invariant subspace* of  $H$ , and it is exactly what is needed in the linear quadratic and  $H_\infty$  control problems [17, 23]. Note that to compute the stable invariant subspace it is not strictly necessary that  $T$  be in quasi triangular form; it can be block triangular with some blocks bigger than  $2 \times 2$ , and it can even be full. All that is needed is that the eigenvalues of  $T$  lie in the left half plane.

The third problem is the decision whether the Hamiltonian matrix has eigenvalues on the imaginary axis or not. This problem is required in the  $H_\infty$  control problem [2, 23] and in the context of passivation algorithms [10, 20]. The critical situation is in this case if in the Hamiltonian real Schur form eigenvalues

are detected that are so near to the imaginary axis that they could have been moved there by small perturbations that are on the order of the error committed by the computation of the Hamiltonian real Schur form. It will depend on the application, what the best strategy is to deal with this case.

## 4 Description of the algorithm

Consider a Hamiltonian matrix  $H$  that is to be transformed to Hamiltonian real Schur form if it exists or to the form (3). Our algorithm consists of a sequence of orthogonal symplectic similarity transformations

$$H \leftarrow \hat{Q}^T H \hat{Q}.$$

Each of these is accompanied by an update

$$Q \leftarrow Q \hat{Q},$$

where  $Q$  was initially  $I_{2n}$ . In the end,  $H$  will have been transformed either to a Hamiltonian real Schur form

$$H_{\text{final}} = \begin{bmatrix} T & C \\ 0 & -T^T \end{bmatrix},$$

where  $T$  is quasi triangular or block triangular, depending on the application, or  $H_{\text{final}}$  is in the form (3). We have

$$H = Q H_{\text{final}} Q^T. \quad (5)$$

The first step of the algorithm, which is the same as the first step of the CLM method, is to compute a symplectic  $URV$  decomposition of  $H$ . We recall the following result [5, 13, 22].

**Theorem 3** (*Symplectic URV decomposition*) *Let  $H \in \mathbb{R}^{2n \times 2n}$  be Hamiltonian. Then there exist orthogonal symplectic  $U, V \in \mathbb{R}^{2n \times 2n}$ , upper-triangular  $T \in \mathbb{R}^{n \times n}$ , quasitriangular  $S \in \mathbb{R}^{n \times n}$ , and  $G \in \mathbb{R}^{n \times n}$  such that*

$$H = U R_1 V^T = V R_2 U^T,$$

where

$$R_1 = \begin{bmatrix} S & G \\ 0 & T^T \end{bmatrix} \quad \text{and} \quad R_2 = \begin{bmatrix} -T & G^T \\ 0 & -S^T \end{bmatrix}.$$

The eigenvalues of  $H$  are the square roots of the eigenvalues of the quasitriangular matrix  $-ST$ .

A symplectic  $URV$  decomposition can be computed by a backward stable algorithm in  $O(n^3)$  flops, and it is implemented in HAPACK [3].

The next step is to partition the eigenvalues of  $H^2$ , which are known once we have the symplectic  $URV$  decomposition. For each pair of eigenvalues  $(\lambda, -\lambda)$  of

$H$ ,  $H^2$  has a single eigenvalue  $\lambda^2$  of multiplicity two. Thus we are partitioning a set of  $n$  numbers. The numbers we actually work with are the eigenvalues of  $-ST$ , which are the eigenvalues of  $H^2$ .

The blocks of eigenvalues do not all need to be of the same size. They need to be big enough that all eigenvalues in a tight cluster are in the same block. Every pair of complex conjugate eigenvalues needs to be in the same block. We may also specify a minimum block size in the interest of computational efficiency. It is not important that all eigenvalues in a block are close together. What matters is that the eigenvalues in each block are reasonably well separated from the eigenvalues in the other blocks. In the interest of satisfying this requirement, it will occasionally happen that all  $n$  eigenvalues lie in a single block.

Clusters are identified as follows. With each eigenvalue  $\mu$  of  $H^2$  we associate an open ball  $\mathcal{B}_\mu$  centered at  $\mu$  with radius proportional to its condition number as an eigenvalue of  $-ST$ . To be precise, we take the radius to be  $10\|S\|_F\|T\|_F\kappa\epsilon$ , where  $\kappa$  is the eigenvalue's condition number, see e.g. [9, 22], and  $\epsilon = 2.22 \times 10^{-16}$  is the "machine epsilon". Let  $\mathcal{B} = \bigcup \mathcal{B}_\mu$ . Then each connected component of  $\mathcal{B}$  defines a cluster, except that whenever there are two clusters such that each has eigenvalues that are the complex conjugates of the eigenvalues of the other cluster, those two clusters are combined to form a single cluster. Eigenvalues in the same cluster are always placed in the same block.

It is at this point that purely imaginary eigenvalues can be identified, as these correspond to real, non-positive eigenvalues of  $-ST$ . If there are any such eigenvalues, they are placed in a block by themselves.

Let us suppose there are  $s$  blocks of eigenvalues and denote the  $i$ th block by  $\Lambda_i$ . We require that the eigenvalues in each block lie in adjacent positions on the main diagonal of  $-ST$ . That is, we require that

$$-ST = B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1s} \\ & B_{22} & & B_{2s} \\ & & \ddots & \vdots \\ & & & B_{ss} \end{bmatrix}, \quad (6)$$

where the spectrum of  $B_{ii}$  is  $\Lambda_i$ ,  $i = 1, \dots, s$ . If there is a block of purely imaginary eigenvalues, that block is placed last. This blocking is achieved by a reordering of the eigenvalues in the symplectic  $URV$  decomposition, see [7].

Once we have a symplectic  $URV$  decomposition with eigenvalues in the desired order, our initial transformation is

$$H \leftarrow U^T H U,$$

where  $U$  is the orthogonal symplectic matrix of Theorem 3. This new  $H$  is a full Hamiltonian matrix that has no overt additional structure. However, when one looks at its square, one finds that

$$H^2 = R_1 R_2 = \begin{bmatrix} -ST & N \\ 0 & (-ST)^T \end{bmatrix} = \begin{bmatrix} B & N \\ 0 & B^T \end{bmatrix},$$

where  $N$  is skew-symmetric and the form of  $B$  is given by (6). The block triangular form of  $H^2$  is crucial to the functioning of the algorithm. All of the transformations of the algorithm are designed to preserve this form of  $H^2$ . Note, however, that we never actually compute  $H^2$ . One additional point is that the blocks  $B_{ii}$  in (6) are themselves quasi-triangular, but we do not propose to use or preserve this structure. By sacrificing this much structure, we are able to process clusters of eigenvalues accurately.

Let  $k$  ( $= k_1$ ) denote the dimension of the matrix  $B_{11}$ . This is the size of the first block of eigenvalues. Let  $E_k \in \mathbb{R}^{2n \times k}$  denote the submatrix of  $I_{2n}$  consisting of the first  $k$  columns. Then the form of  $H^2$  implies that  $\mathcal{R}(E_k)$  is invariant under  $H^2$ , as  $H^2 E_k = E_k B_{11}$ . Therefore,  $\mathcal{R}([ E_k \ HE_k ])$  is invariant under  $H$ , i.e. we have

$$H [ E_k \ HE_k ] = [ E_k \ HE_k ] \begin{bmatrix} 0 & B_{11} \\ I & 0 \end{bmatrix}. \quad (7)$$

Generically, the dimension of  $\mathcal{R}([ E_k \ HE_k ])$  will be  $2k$ , and we will assume for now that it is. Exceptional cases and other details will be discussed in Section 6. The  $2k$  eigenvalues associated with this invariant subspace are therefore the eigenvalues of

$$\begin{bmatrix} 0 & B_{11} \\ I & 0 \end{bmatrix},$$

which are the  $2k$  square roots of the  $k$  eigenvalues in the first block, i.e. the square roots of the elements of  $\Lambda_1$ .

The next step of the algorithm is to pick out a  $k$ -dimensional isotropic invariant subspace of  $\mathcal{R}([ E_k \ HE_k ])$ . By Theorem 1 we see that we can do this by picking out a subspace  $\mathcal{S}$  such that for each eigenvalue  $\lambda$  associated with  $\mathcal{S}$ ,  $-\lambda$  is not associated with  $\mathcal{S}$ . For example, if there are no purely imaginary eigenvalues, we can take  $\mathcal{S}$  to be the subspace associated with the  $k$  eigenvalues with negative real part.

We proceed as follows. Compute a  $QR$  decomposition

$$[ E_k \ HE_k ] = QR, \quad (8)$$

where  $Q \in \mathbb{R}^{2n \times 2k}$  has orthonormal columns. This gives an orthonormal basis for the space. Let  $G = Q^T H Q$ . Use the  $QR$  algorithm to compute the Schur decomposition  $G = \tilde{Q} \tilde{T} \tilde{Q}^T$ , where  $\tilde{Q}$  is orthogonal and  $\tilde{T}$  is quasi-triangular. In this step we include a sorting operation that moves the eigenvalues that we wish to retain (e.g. the eigenvalues in the left half plane) to the upper left of  $\tilde{T}$ . Let  $\tilde{Q} = [ \tilde{Q}_1 \ \tilde{Q}_2 ]$ , where  $\tilde{Q}_1 \in \mathbb{R}^{2k \times k}$ . Since  $G \tilde{Q}_1 = \tilde{Q}_1 \tilde{T}_{11}$ , where  $\tilde{T}_{11}$  is the upper-left  $k \times k$  submatrix of  $\tilde{T}$ , the columns of  $\tilde{Q}_1$  form an orthonormal basis of the invariant subspace of  $G$  associated with the eigenvalues that are to be retained. Let

$$X = Q \tilde{Q}_1.$$



Then  $X$  is an orthonormal basis of the desired invariant subspace of  $H$ . We have

$$HX = XF, \quad (9)$$

where  $F = \tilde{T}_{11} \in \mathbb{R}^{k \times k}$ .

We will obtain a deflation by transforming  $X$  to  $E_k$ , the matrix consisting of the first  $k$  columns of the identity matrix. That is, we will find an orthogonal symplectic  $S$  such that  $S^T X = E_k$  and make the transformation  $H \leftarrow S^T H S$ . We will build and apply  $S$  by stages in such a way that the block triangular form of  $H^2$  is preserved. Notice that the isotropy of  $\mathcal{R}(X)$  is crucial. Since the equation  $S^T X = E_k$  is equivalent to  $S E_k = X$ ,  $\mathcal{R}(E_k)$  is isotropic, and the symplectic matrix  $S$  preserves isotropy. Thus we expect to have to exploit isotropy at some point in the reduction.

Let  $k_1, \dots, k_s$  denote the sizes of the main-diagonal blocks in (6). These are exactly the cardinalities of the sets  $\Lambda_1, \dots, \Lambda_s$  of eigenvalues of  $H^2$ . Note that  $k = k_1$ . Partition  $X$  conformably with these blocks as

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_s \\ X_{s+1} \\ \vdots \\ X_{2s} \end{bmatrix},$$

where  $X_j$  and  $X_{s+j}$  are  $k_j \times k$ ,  $j = 1, \dots, s$ .

The first stage of the reduction operates on blocks  $X_{s+1}$  and  $X_{s+2}$ . Let  $\check{S}$  be an orthogonal matrix of dimension  $k_1 + k_2$  such that

$$\check{S}^T \begin{bmatrix} X_{s+1} \\ X_{s+2} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{X}_{s+2} \end{bmatrix}, \quad (10)$$

where the zero block occupies  $k_2$  rows and  $\tilde{X}_{s+2}$  is  $k_1 \times k_1$ . This can be achieved by an ‘‘upside down’’  $QR$  decomposition: Let  $\tilde{F}$  denote the *flip matrix*, the matrix obtained by reversing the columns of the identity matrix. If we take a

$QR$  decomposition  $\begin{bmatrix} X_{s+1} \\ X_{s+2} \end{bmatrix} = QR$ , then note that  $QR = (Q\tilde{F})(\tilde{F}R)$ , we see

that we can take  $\check{S} = Q\tilde{F}$  and  $\begin{bmatrix} 0 \\ \tilde{X}_{s+2} \end{bmatrix} = \tilde{F}R$ .

With  $\hat{S} = \text{diag}\{\check{S}, I\} \in \mathbb{R}^{n \times n}$  and

$$\tilde{S} = \begin{bmatrix} \hat{S} & 0 \\ 0 & \hat{S} \end{bmatrix},$$

we have that  $\tilde{S}$  is orthogonal and symplectic.

Performing the updates

$$X \leftarrow \tilde{S}^T X \quad \text{and} \quad H \leftarrow \tilde{S}^T H \tilde{S},$$

we see that equation (9) continues to hold for the new  $X$  and  $H$ . But the new  $X$  has a  $k_2 \times k_1$  block of zeros. We adjust the partition accordingly, i.e. we have

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_s \\ 0 \\ X_{s+2} \\ \vdots \\ X_{2s} \end{bmatrix},$$

where  $X_{s+2}$  and  $X_2$  are  $k_1 \times k_1$  and  $X_{s+1} (= 0)$  and  $X_1$  are  $k_2 \times k_1$ . All other blocks are unchanged by this transformation.

The second stage is identical to the first, except that it operates on the blocks  $X_{s+2}$  and  $X_{s+3}$ , transforming

$$\begin{bmatrix} X_{s+2} \\ X_{s+3} \end{bmatrix} \text{ to } \begin{bmatrix} 0 \\ \tilde{X}_{s+3} \end{bmatrix},$$

where the zero block has  $k_3$  rows and  $\tilde{X}_{s+3}$  is  $k_1 \times k_1$ . Building a transforming matrix and performing a transformation analogous to that of the first stage, we obtain a new  $H$  and  $X$  that continue to satisfy (9). This new  $X$  has  $k_2 + k_3$  rows of zeros.

Each subsequent stage introduces another block of zeros into  $X$ . After  $s - 1$  stages, the transformed  $X$  has the form

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_{s-1} \\ X_s \\ 0 \\ \vdots \\ 0 \\ X_{2s} \end{bmatrix}, \quad (11)$$

where  $X_s$  and  $X_{2s}$  are  $k_1 \times k_1$ , and the other blocks  $X_j$  are  $k_{j+1} \times k_1$ .

### Exploiting Isotropy in stage $s$

Stage  $s$  is special. Here we operate on blocks  $X_s$  and  $X_{2s}$  to transform  $X_{2s}$  to zero. Specifically, we produce an orthogonal symplectic matrix  $\begin{bmatrix} \tilde{Q}_1 & -\tilde{Q}_2 \\ \tilde{Q}_2 & \tilde{Q}_1 \end{bmatrix}$  such that

$$\begin{bmatrix} \tilde{Q}_1 & -\tilde{Q}_2 \\ \tilde{Q}_2 & \tilde{Q}_1 \end{bmatrix}^T \begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} = \begin{bmatrix} \tilde{X}_s \\ 0 \end{bmatrix}. \quad (12)$$

We describe two ways of doing this. The original space  $\mathcal{R}(X)$  was isotropic, and so is the transformed  $\mathcal{R}(X)$ , because symplectic transformations preserve isotropy, see e.g. [6]. Because of the zero blocks in  $X$ , we have  $0 = X^T J X = X_s^T X_{2s} - X_{2s}^T X_s$ , which implies that the columns of  $\begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} \in \mathbb{R}^{2k \times k}$  also span an isotropic space. However, the columns are not orthonormal. To get an orthonormal basis, we take a condensed  $QR$  decomposition

$$\begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} = \begin{bmatrix} \check{Q}_1 \\ \check{Q}_2 \end{bmatrix} R, \quad (13)$$

where  $R$  is  $k \times k$ . Then

$$\begin{bmatrix} \check{Q}_1 & -\check{Q}_2 \\ \check{Q}_2 & \check{Q}_1 \end{bmatrix} \quad (14)$$

is symplectic and orthogonal and satisfies (12) with  $\tilde{X}_s = R$ .

Let  $\hat{Q}_1 = \text{diag}\{I, \hat{Q}_1\} \in \mathbb{R}^n$ ,  $\hat{Q}_2 = \text{diag}\{I, \hat{Q}_2\} \in \mathbb{R}^n$ , and

$$\tilde{Q} = \begin{bmatrix} \hat{Q}_1 & -\hat{Q}_2 \\ \hat{Q}_2 & \hat{Q}_1 \end{bmatrix}. \quad (15)$$

Performing the updates

$$X \leftarrow \tilde{Q}^T X \quad \text{and} \quad H \leftarrow \tilde{Q}^T H \tilde{Q}, \quad (16)$$

equation (9) continues to hold for the new  $X$  and  $H$ , and the updated  $X$  has  $X_{2s} = 0$ .

This method of annihilating  $X_{2s}$  has the following weakness. In cases where  $\max\{\|X_s\|, \|X_{2s}\|\}$  is very small, the  $QR$  decomposition (13) amplifies errors. Even if the space is isotropic to working precision, the isotropy can be lost at this stage, and the corresponding transformation will not be truly symplectic. Therefore this approach can fail when  $\max\{\|X_s\|, \|X_{2s}\|\}$  is small.

We employ the following remedy. Before making the updates (16), we check that the space spanned by the columns of  $Y = \begin{bmatrix} \hat{Q}_1 \\ \hat{Q}_2 \end{bmatrix}$  really is isotropic. If it fails to be below our isotropy tolerance (which is specified precisely in Section 6), we abandon  $Y$  and use the following alternative procedure.

Initially  $\tilde{F}X_{2s}$  is upper triangular, so that, in the case  $k = 4$  for example,

$$\begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} = \frac{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}{\begin{matrix} * \\ * & * \\ * & * & * \\ * & * & * & * \end{matrix}}. \quad (17)$$

The transformation  $X \rightarrow \tilde{Q}^T X$  described in the first method is replaced by a sequence of symplectic Givens rotations. Thus in the second method the total transformation matrix becomes a product of transformations that are both symplectic and orthogonal to working precision. Thus the overall transformation is symplectic (and orthogonal) to working precision.

The first step of the reduction applies a (symplectic) rotation to rows  $k$  and  $2k$  of (17) (which correspond to rows  $n$  and  $2n$  in the big matrix) to annihilate the  $(k, 1)$  entry of  $X_{2s}$ . The second step applies a symplectic double rotation that acts on rows  $2k-1$  and  $2k$  and on rows  $k-1$  and  $k$  to annihilate the  $(k-1, 2)$  entry of  $X_{2s}$ . By isotropy of the first two columns, this transformation must also annihilate the  $(k, 1)$  entry of  $X_s$ . Thus this transformation introduces two zeros simultaneously. The next step applies another symplectic double rotation. This one acts on rows  $2k-2$  and  $2k-1$  and also on rows  $k-2$  and  $k-1$  to annihilate the  $(k-2, 3)$  entry of  $X_{2s}$ . By isotropy it must also annihilate the  $(k-1, 1)$  entry of  $X_s$ . We continue in this way, moving up the antidiagonal of  $X_{2s}$ , applying symplectic double rotations that annihilate antidiagonal entries of  $X_{2s}$  while simultaneously producing zeros in the first column of  $X_s$ . After  $k$  double rotations, we have

$$\begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} = \left[ \begin{array}{cccc|cccc} * & * & * & * & & & & \\ & * & * & * & & & & \\ & & * & * & & & & \\ & & & * & * & * & & \\ & & & & & & & * \\ & & & & & & & * \\ & & & & & & & * \\ & & & & & & & * \end{array} \right].$$

This is the end of the first major step. The second major step annihilates the next antidiagonal of  $X_{2s}$  by a procedure entirely analogous to the first step. First a symplectic rotation acting on rows  $k$  and  $2k$  eliminates the  $(k, 2)$  entry of  $X_{2s}$ . Then a sequence of double rotations is applied to eliminate the entries  $(k-1, 3)$ ,  $(k-2, 4)$ , and so on, in  $X_{2s}$ , while simultaneously annihilating entries in the second column of  $X_s$ . At the end of the second major step we have

$$\begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} = \left[ \begin{array}{cccc|cccc} * & * & * & * & & & & \\ & * & * & * & & & & \\ & & * & * & & & & \\ & & & * & * & * & & \\ & & & & & & & * \\ & & & & & & & * \\ & & & & & & & * \\ & & & & & & & * \end{array} \right].$$

After  $k$  major steps the procedure is finished.  $X_{2s}$  has been transformed to zero, and  $X_s$  has been transformed to upper-triangular form.

## The final stages of the transformation

The next stage operates on blocks  $X_{s-1} \in \mathbb{R}^{k_s \times k_1}$  and  $X_s \in \mathbb{R}^{k_1 \times k_1}$ . Let  $\check{S}$  be an orthogonal matrix of dimension  $k_s + k_1$  such that

$$\check{S}^T \begin{bmatrix} X_{s-1} \\ X_s \end{bmatrix} = \begin{bmatrix} \tilde{X}_{s-1} \\ 0 \end{bmatrix},$$

where  $\tilde{X}_s$  is  $k_1 \times k_1$ , and the block of zeros occupies  $k_s$  rows. This can be achieved by an ordinary  $QR$  decomposition. Let  $\hat{S} = \text{diag}\{I, \check{S}\} \in \mathbb{R}^{n \times n}$  and  $\tilde{S} = \text{diag}\{\hat{S}, \hat{S}\}$ .

Performing the updates

$$X \leftarrow \tilde{S}^T X \quad \text{and} \quad H \leftarrow \tilde{S}^T H \tilde{S},$$

equation (9) continues to hold, and the new  $X$  has the form

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_{s-1} \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where  $X_{s-1}$  is  $k_1 \times k_1$ .

The next stage, which is essentially the same, operates on blocks  $X_{s-2} \in \mathbb{R}^{k_{s-1} \times k_1}$  and  $X_{s-1} \in \mathbb{R}^{k_1 \times k_1}$  and produces another  $k_{s-1}$  rows of zeros.

Each subsequent stage introduces one more block of zeros into  $X$ . After  $2s - 1$  stages we have

$$X = \begin{bmatrix} X_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} = E_k.$$

This form of  $X$  and the equation (9)  $HX = XF$  imply that the transformed  $H$  has the form

$$H = \left[ \begin{array}{cc|cc} A_{11} & A_{12} & G_{11} & G_{12} \\ 0 & A_{22} & G_{21} & G_{22} \\ \hline 0 & 0 & -A_{11}^T & 0 \\ 0 & C_{22} & -A_{12}^T & -A_{22}^T \end{array} \right],$$

where  $A_{11} = F \in \mathbb{R}^{k \times k}$ . The zeros in the first column follow from the equation  $HX = XF$ , and the other zeros are implied by Hamiltonian structure. It is now possible to deflate the problem and work with the Hamiltonian submatrix

$$\begin{bmatrix} A_{22} & G_{22} \\ C_{22} & -A_{22}^T \end{bmatrix}$$

of dimension  $2(n - k)$ . In Section 5 we will show the crucial fact that the transformations outlined above preserve the block-triangular structure of  $H^2$ . The first step of the algorithm relied on this structure to obtain a low-dimensional subspace that was invariant under  $H$ . After the deflation we want to perform another step that is just like the first, and for this we need that the block-triangular structure of  $H^2$  has been kept intact.

If no difficulties are encountered along the way (see Section 6 for a discussion of difficulties), the algorithm terminates after  $s$  steps. If no purely imaginary eigenvalues have been detected, then  $H$  will have been transformed to the form

$$H = \begin{bmatrix} A & G \\ 0 & -A^T \end{bmatrix}, \quad (18)$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1s} \\ 0 & A_{22} & & A_{2s} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & A_{ss} \end{bmatrix}$$

is block triangular. Each  $A_{ii}$  is  $k_i \times k_i$  and has eigenvalues that are square roots of the elements of  $\Lambda_i$ .

If some eigenvalues were found that were either purely imaginary or too close to being purely imaginary, then instead of the block of zeros in (18), there will be a block

$$K = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & K_{ss} \end{bmatrix},$$

where the eigenvalues of the Hamiltonian matrix

$$\begin{bmatrix} A_{ss} & G_{ss} \\ K_{ss} & -A_{ss}^T \end{bmatrix}$$

are on or near the imaginary axis. The code reports whether all of these eigenvalues are all actually on the imaginary axis or merely near the imaginary axis or a mixture of the two types.

## Postprocessing

The further actions taken at this point depend upon the application and which of the different problems discussed in Section 3 we want to solve. Suppose  $H$  has no eigenvalues on or close to the imaginary axis and that we want to compute the stable invariant subspace.

If the isotropic subspace used at each step was chosen so that its eigenvalues are in the left half plane, then  $A_{11}, A_{22}, \dots, A_{ss}$  will contain all of the left-half-plane eigenvalues and no others. In this case we are done. We have accumulated

the transforming matrix  $Q$  of (5), and its first  $n$  columns are an orthonormal basis of the stable invariant subspace.

If some of the  $A_{ii}$  have eigenvalues from the right half plane, or if our task is to compute the Hamiltonian real Schur form, then further processing is necessary. Each of the  $A_{ii}$  can be transformed to quasi-triangular form by the standard unstructured  $QR$  algorithm:  $T_{ii} = U_{ii}^T A_{ii} U_{ii}$ , where  $T_{ii}$  is quasi-triangular and  $U_{ii}$  is orthogonal. Let  $\hat{U} = \text{diag}\{U_{11}, \dots, U_{ss}\}$  and  $U = \text{diag}\{\hat{U}, \hat{U}\}$ . Then a similarity transformation by the orthogonal symplectic matrix  $U$  transforms  $H$  to real Hamiltonian Schur form.

If the order of the eigenvalues is not of interest, we are done. If the eigenvalues need to be reordered, for example to get the left-half-plane eigenvalues to the top in order to extract the stable invariant subspace, one additional sorting step is necessary. This can be accomplished by the sorting routine from HAPACK [3] mentioned earlier.

## Extreme cases

The algorithm we have just described reduces to CLM in the extreme case when the block sizes are made as small as possible, i.e. each real eigenvalue forms a block of size one by itself and each complex conjugate pair forms a block of size two.

At the opposite extreme we have the case  $s = 1$ , in which there is a single block of  $n$  eigenvalues. It is instructive to think about what the algorithm does in this case (provided that a Hamiltonian real Schur form exists.) First of all, it is not necessary to compute the  $QR$  decomposition (8) or the projection matrix  $F = Q^T H Q$ , because in this case  $Q$  is square and the transformation  $F = Q^T H Q$  is just a similarity transformation. Instead of computing the Schur decomposition of  $F$ , we can just compute the Schur decomposition of  $H$  by the  $QR$  algorithm. We then pick out an  $n$ -dimensional isotropic invariant subspace by moving the eigenvalues that we wish to retain to the top of the quasi-triangular matrix. Once the sorting has been done, our invariant subspace  $\mathcal{R}(X)$  is given by the first  $n$  columns of  $Q$  in the Schur factorization  $H = Q T Q^T$ . In this case the reduction of  $X$  consists of only one step, the one in which  $X_s$  and  $X_{2s}$  are modified and  $X_{2s}$  is transformed to zero (we have  $s = 1$ ). In this case the  $QR$  decomposition (13) is not necessary because the columns of  $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$  are already orthonormal. We just take  $\check{Q}_1 = X_1$  and  $\check{Q}_2 = X_2$ .

The matrix (14), which is the same as  $\tilde{Q}$  in (15) in this case, is orthogonal and symplectic because  $\mathcal{R}(X)$  is isotropic. Then the update  $H \leftarrow \tilde{Q}^T H \tilde{Q}$  yields a Hamiltonian block triangular  $H$ .

This technique of using an unstructured method (the standard  $QR$  algorithm) to effect a structured transformation is commonly known as the *Laub trick* [14]. Thus our algorithm just reduces to the Laub trick in the case when all of the eigenvalues are placed in a single block.

## 5 Why the algorithm works

The main task of this section is to show that the transformations outlined in the previous section do not disturb the block-triangular structure of  $H^2$ . We will also show that eigenvalue swapping takes place at each step. Consider the point in the computation at which the matrix  $X \in \mathbb{R}^{2n \times k}$  satisfying (9) has just been computed. This matrix has orthonormal columns and spans an isotropic invariant subspace. Of course this space is also invariant under  $H^2$ , i.e.

$$H^2 X = X F^2. \quad (19)$$

Making the partition

$$X = \begin{bmatrix} Y \\ Z \end{bmatrix}$$

and recalling that  $H^2$  has the form

$$\begin{bmatrix} B & N \\ 0 & B^T \end{bmatrix},$$

we see that

$$B^T Z = Z F^2.$$

Since  $B$  has the block-triangular form (6), the first two blocks of this equation yield

$$\begin{bmatrix} B_{11}^T & 0 \\ B_{12}^T & B_{22}^T \end{bmatrix} \begin{bmatrix} X_{s+1} \\ X_{s+2} \end{bmatrix} = \begin{bmatrix} X_{s+1} \\ X_{s+2} \end{bmatrix} F^2, \quad (20)$$

and the first block by itself gives

$$B_{11}^T X_{s+1} = X_{s+1} F^2. \quad (21)$$

We now make the assumption that the square matrix  $X_{s+1}$  is nonsingular. In Section 6 we will address the question of what to do if this assumption is violated or nearly violated.

Since  $X_{s+1}$  is nonsingular, (21) shows that  $F^2$  is similar to  $B_{11}^T$ . Thus the spectrum of  $F^2$  is just  $\Lambda_1$ , the set of eigenvalues of the first block.

We transform  $X$  to  $E_k$  by stages. The first stage tells the whole story, more or less. The transformation (10), together with the corresponding transformation of  $H$  implies a transformation of (20) to

$$\begin{bmatrix} \tilde{B}_{11}^T & \tilde{B}_{21}^T \\ \tilde{B}_{12}^T & \tilde{B}_{22}^T \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{X}_{s+2} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{X}_{s+2} \end{bmatrix} F^2. \quad (22)$$

Recall that  $\tilde{X}_{s+2}$  is  $k_1 \times k_1$ . We partition  $\tilde{B}$  conformably with  $\tilde{X}$ , i.e.  $\tilde{B}_{11}$  is  $k_2 \times k_2$ , and  $\tilde{B}_{22}$  is  $k_1 \times k_1$ . Since  $X_{s+1}$  is nonsingular,  $\tilde{X}_{s+2}$  is certainly also nonsingular. Indeed,  $\tilde{X}_{s+2}^T \tilde{X}_{s+2} = X_{s+1}^T X_{s+1} + X_{s+2}^T X_{s+2}$ , so the smallest singular value of  $\tilde{X}_{s+2}$  is no smaller than that of  $X_{s+1}$ . Looking at the first equation of (22), we find that  $\tilde{B}_{21}^T \tilde{X}_{s+2} = 0$ . Since  $\tilde{X}_{s+2}$  is nonsingular, it



also follows that  $\tilde{B}_{21}^T = 0$ . This shows that the block triangular form of  $B$  has been preserved. Notice further that the second equation of (22) yields  $\tilde{B}_{22}^T \tilde{X}_{s+2} = \tilde{X}_{s+2} F^2$ . Thus  $\tilde{B}_{22}^T$  is similar to  $F^2$ , which shows that the spectrum of  $B_{22}$  is  $\Lambda_1$ . It follows that the spectrum of  $\tilde{B}_{11}$  is  $\Lambda_2$ . The blocks of eigenvalues have been swapped!

The second stage of the reduction works the same as the first. If we now drop the tildes for notational simplicity and look at the second and third blocks of the equation  $B^T Z = Z F^2$ , taking into account that the first block of  $Z$  is now zero, we obtain

$$\begin{bmatrix} B_{22}^T & 0 \\ B_{23}^T & B_{33}^T \end{bmatrix} \begin{bmatrix} X_{s+2} \\ X_{s+3} \end{bmatrix} = \begin{bmatrix} X_{s+2} \\ X_{s+3} \end{bmatrix} F^2, \quad (23)$$

The spectra of  $B_{22}$  and  $B_{33}$  are  $\Lambda_1$  and  $\Lambda_3$ , respectively, and  $X_{s+2}$  is nonsingular. The second transformation turns (23) into

$$\begin{bmatrix} \tilde{B}_{22}^T & \tilde{B}_{32}^T \\ \tilde{B}_{23}^T & \tilde{B}_{33}^T \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{X}_{s+3} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{X}_{s+3} \end{bmatrix} F^2,$$

where  $\tilde{X}_{s+3}$  and  $\tilde{B}_{33}^T$  are  $k_1 \times k_1$  and  $\tilde{B}_{22}^T$  is  $k_3 \times k_3$ . Clearly  $\tilde{X}_{s+3}$  is nonsingular (and its smallest singular value is no smaller than that of  $X_{s+2}$ ). We then deduce immediately that  $\tilde{B}_{32}^T = 0$  and  $\tilde{B}_{23}^T$  is similar to  $F^2$ . Thus the spectra of  $\tilde{B}_{33}$  and  $\tilde{B}_{22}$  are  $\Lambda_3$  and  $\Lambda_1$ , respectively. Again the eigenvalues have been interchanged.

After  $s-1$  stages,  $X$  will be transformed to the form (11). The block triangular form of  $B$  is preserved, but (leaving off the tildes)  $B_{ii}$  is  $k_{i+1} \times k_{i+1}$  and has spectrum  $\Lambda_{i+1}$ ,  $i = 1, \dots, s-1$ , and  $B_{ss}$  is  $k_1 \times k_1$  and has spectrum  $\Lambda_1$ . Extracting block equations  $s$  and  $2s$  from the (transformed) equation  $H^2 X = X F^2$ , we have

$$\begin{bmatrix} B_{ss} & N_{ss} \\ 0 & B_{ss}^T \end{bmatrix} \begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} = \begin{bmatrix} X_s \\ X_{2s} \end{bmatrix} F^2. \quad (24)$$

$X_{2s}$  is nonsingular. The transformation of the  $s$ th stage turns (24) into

$$\begin{bmatrix} \tilde{B}_{ss} & \tilde{N}_{ss} \\ \tilde{K}_{ss} & \tilde{B}_{ss}^T \end{bmatrix} \begin{bmatrix} \tilde{X}_s \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{X}_s \\ 0 \end{bmatrix} F^2. \quad (25)$$

It is easy to check that this transformation produces no fill in anywhere in  $H^2$ , except possibly in the block that we have labelled  $\tilde{K}_{ss}$ . Certainly  $\tilde{X}_s$  is nonsingular. It then follows from the second equation of (25) that  $\tilde{K}_{ss} = 0$ . The first equation shows that  $\tilde{B}_{ss}$  is similar to  $F^2$ , so its spectrum is  $\Lambda_1$ , as expected.

Dropping the tildes again and moving on to the next stage, we now have  $H^2 X = X F^2$ , where the bottom half of  $X$  consists entirely of zeros. Writing  $X = \begin{bmatrix} Y \\ 0 \end{bmatrix}$ , we have  $BY = Y F^2$ . The bottom two blocks of this equation give

$$\begin{bmatrix} B_{s-1,s-1} & B_{s-1,s} \\ 0 & B_{ss} \end{bmatrix} \begin{bmatrix} X_{s-1} \\ X_s \end{bmatrix} = \begin{bmatrix} X_{s-1} \\ X_s \end{bmatrix} F^2. \quad (26)$$

$X_s$  and  $B_{ss}$  are  $k_1 \times k_1$ , and  $B_{s-1,s-1}$  is  $k_s \times k_s$ . The spectra of  $B_{ss}$  and  $B_{s-1,s-1}$  are  $\Lambda_1$  and  $\Lambda_s$ , respectively.  $X_s$  is nonsingular. The next transformation turns (26) into

$$\begin{bmatrix} \tilde{B}_{s-1,s-1} & \tilde{B}_{s-1,s} \\ \tilde{B}_{s,s-1} & \tilde{B}_{ss} \end{bmatrix} \begin{bmatrix} \tilde{X}_{s-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{X}_{s-1} \\ 0 \end{bmatrix} F^2. \quad (27)$$

$\tilde{X}_{s-1}$  is  $k_1 \times k_1$  and certainly nonsingular. The second equation shows that  $\tilde{B}_{s,s-1} = 0$ . The first shows that  $\tilde{B}_{s-1,s-1}$  is similar to  $F^2$ . Thus the spectrum of  $\tilde{B}_{s-1,s-1}$  is  $\Lambda_1$ , and that of  $B_{s,s}$  is  $\Lambda_s$ .

After  $s-2$  more transformations, we are done. Leaving off the tildes, we have  $H^2 X = X F^2$ , where  $X = E_k$ . The block triangular structure of  $B$  has been preserved. Each main-diagonal block  $B_{ii}$  is  $k_i \times k_i$  and has spectrum  $\Lambda_i$ . We are back where we started with  $H^2$ . Meanwhile, however,  $H$  has been transformed into a form that allows a deflation.

## 6 Computational details

We have already described our method of partitioning the eigenvalues into blocks. However, this is only a tentative blocking, and we are willing modify it on the fly if necessary. Certainly we have no objection to combining two blocks into a single block if needed. As we have already remarked, it will sometimes be necessary to include all of the eigenvalues in a single block.

The first potential difficulty is that the space  $\mathcal{R}(\begin{bmatrix} E_k & H E_k \end{bmatrix})$  (see (7)) may have dimension less than  $2k$ . The most extreme case of this is when  $\mathcal{R}(E_k)$  already happens to be invariant under  $H$ . To deal with this possibility, write

$$\begin{bmatrix} E_k & H E_k \end{bmatrix} = \begin{bmatrix} I_k & H_{11} \\ 0 & H_{21} \end{bmatrix},$$

and notice that the space spanned by the columns of this matrix is the same as the space spanned by the columns of

$$\begin{bmatrix} I_k & 0 \\ 0 & H_{21} \end{bmatrix}.$$

Now take a  $QR$  decomposition with column pivoting  $H_{21} = URP$ , where  $U$  has orthonormal columns,  $R$  is  $k \times k$  and upper triangular, and  $P$  is a  $k \times k$  permutation matrix. This is inexpensive if  $k \ll n$ . Alternatively an SVD or other rank-revealing decomposition can be used, see [9].

Then the columns of

$$Q = \begin{bmatrix} I_k & 0 \\ 0 & U \end{bmatrix} \quad (28)$$

are an orthonormal basis of  $\mathcal{R}(\begin{bmatrix} E_k & H E_k \end{bmatrix})$ , provided that  $H_{21}$  has full rank. If  $H_{21}$  is rank deficient (or nearly so), the leading columns of (28) are a basis for  $\mathcal{R}(\begin{bmatrix} E_k & H E_k \end{bmatrix})$ , and the extraneous columns are at the end. We use the matrix  $Q$  from (28) instead of the matrix from the  $QR$  decomposition (8).

The next step is to pick out an isotropic invariant subspace of  $\mathcal{R}(Q)$ . To this end we compute the matrix  $G = Q^T H Q$  and its real Schur decomposition  $G = \tilde{Q} \tilde{T} \tilde{Q}^T$ , as prescribed in Section 4. This includes a sorting step in which we move the eigenvalues that we wish to retain to the top of  $\tilde{T}$ . In some applications we prefer to keep the eigenvalues with negative real part, but in practice this is not always an option. Consider an extreme special case in which  $\mathcal{R}(\begin{bmatrix} E_k & H E_k \end{bmatrix}) = \mathcal{R}(E_k)$  and has dimension  $k$ . Suppose further that the eigenvalues of  $H$  associated with the invariant subspace  $\mathcal{R}(E_k)$  all have positive real part. In this case we have no choice but to work with eigenvalues with positive real part.

Because of this and similar examples, the best practical course of action is simply to take the eigenvalues that the algorithm provides and sort them later if necessary. Thus our algorithm acts as follows. First the Schur decomposition with no reordering is computed. We would be happy to take the top  $k$  eigenvalues as the ones that we will retain, but we want to make sure that the space so produced is isotropic. Recall from Theorem 1 that we can ensure isotropy by not selecting any opposite pair  $(\lambda, -\lambda)$ . Therefore we begin by selecting the top eigenvalue. Then each subsequent eigenvalue is selected if it is not (equal to or nearly equal to) the opposite of any eigenvalue that has already been selected. In this way we move through the entire set of eigenvalues of  $T$ , selecting some and rejecting others. We then reorder the real Schur decomposition so that the selected eigenvalues are moved to the top.

The subspace built with the selected eigenvalues should be invariant in principle, but in practice we must check whether it really is invariant within a specified tolerance. We will almost never have an invariance failure on the first step, as the symplectic  $URV$  decomposition is fresh, and  $H^2$  has the right form to guarantee that the space  $\mathcal{R}(\begin{bmatrix} E_k & H E_k \end{bmatrix})$  is essentially invariant. Invariance of the chosen isotropic subspace then follows. However, after a few steps and a few deflations, it may be that roundoff errors cause the invariance to be lost.

To check the invariance of  $\mathcal{S} = \mathcal{R}(X)$ , we compute  $G = X^T H X$  and a matrix of residuals  $|HX - XG|$ . If all entries of this matrix are less than  $100\sqrt{n}\|H\|_F \epsilon$ , where  $\epsilon \approx 2.22 \times 10^{-16}$  is the ‘‘machine epsilon’’, we accept the subspace as invariant. We also check the isotropy by computing the isotropy residual matrix  $|X^T J X|$ . If all entries of this matrix are under  $100\sqrt{n}\epsilon$ , the space is judged to be isotropic. If either of these tests is failed, action is taken.

Our current codes can be run in either of two modes. In mode 1, if the subspace fails to be either invariant or isotropic, the normal action is to perform a new symplectic  $URV$  decomposition on the remaining submatrix. New clusters and blocks are computed as well.

In the rare cases when we have a failure even though there have been no deflations since the last symplectic  $URV$  decomposition was computed (so that computing a new one would not help), we merge the current block with the next one and try again with the bigger block.

In mode 2, if we get a failure of either invariance or isotropy, we first look for a smaller dimensional subspace by dropping some columns from  $X$  until we

get a subspace that is both invariant and isotropic to the specified precision. We then work with a smaller block, merging the rejected columns with the next block.

In the event that we are unable to get an isotropic invariant subspace of any dimension, we proceed as in mode 1, either computing a new symplectic  $URV$  decomposition or combining two blocks.

An advantage of mode 2 is that it tends to decrease the total number of symplectic  $URV$  decompositions that have to be performed in the course of entire computation. A potential disadvantage is that decreasing the block size can result in splitting up a cluster of eigenvalues.

The nature of the end phase of the algorithm depends upon whether or not any purely imaginary eigenvalues were detected. First let us suppose that no purely imaginary eigenvalues were found. Then the final block is processed by the Laub trick: A real Schur form of the remaining  $2k_s \times 2k_s$  submatrix is computed, and the left-half-plane eigenvalues are moved to the top. Then the first  $k_s$  columns of the transforming matrix span the isotropic subspace that we use for the final transformation of  $H$ .

The space is automatically invariant, so we do not need to test for this. However, it may fail to be isotropic in practice. We therefore test for isotropy. In case of failure, we do not complete the last transformation. We return the form (3), where

$$\begin{bmatrix} T_{22} & G_{22} \\ C_{22} & -T_{22}^T \end{bmatrix}$$

is the final, unresolved block. The isotropy failure could result from some of the eigenvalues being too close to the imaginary axis.

If the isotropy test is passed, we perform the transformation of the final block to get block Hamiltonian Schur form. If we want the real Hamiltonian Schur form, each of the blocks can be reduced to real Schur form individually by the standard  $QR$  algorithm. This step respects Hamiltonian structure, as whatever transformations are done to blocks of  $T$  in

$$\begin{bmatrix} T & G \\ 0 & -T^T \end{bmatrix},$$

the same transformations are applied to the blocks of  $-T^T$ . The overall transformations are both orthogonal and symplectic. If the eigenvalues need further sorting, this can be done by an algorithm that is provided in HAPACK [3].

In cases in which purely imaginary eigenvalues have been detected, the best result we can get is the form (3), in which the block

$$\begin{bmatrix} T_{22} & G_{22} \\ C_{22} & -T_{22}^T \end{bmatrix} \tag{29}$$

contains the purely imaginary eigenvalues. The worst that can happen is that while processing the final non-purely-imaginary block, we are unable to obtain a subspace that is both invariant and isotropic, even though a new symplectic

$URV$  decomposition has just been computed. This will not normally happen unless there are eigenvalues very close to the imaginary axis. In this case we are unable to process the block, so we end with a form (3) in which the block (29) contains both purely imaginary eigenvalues and eigenvalues that are nearly purely imaginary.

One other question needs to be discussed. In the elimination process the leading block ( $X_{s+1}$  in (10)) was assumed to be nonsingular in Section 5, in which we explained why the method works. What should we do in situations in which  $X_{s+1}$  is singular or nearly singular? We experimented with a strategy in which the current block is merged with the next block whenever this occurs, but in the end we decided to take no special action; we simply proceed with the elimination. This causes no problem with the current elimination, but it can cause problems downstream. The preservation of the block-triangular form of  $H^2$  depends upon the nonsingularity of  $X_{s+1}$ , so there is the danger that the special form of  $H^2$  will be lost. This would necessitate the computation of a new symplectic  $URV$  decomposition to restore the form of  $H^2$ . On the other hand, there are benign situations in which  $X_{s+1}$  is singular. For example, suppose

$$X = \begin{bmatrix} X_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

in which  $X_{s+1} = 0$ . In this situation  $H$  is ready for deflation with no transformation at all. If we take some special action because  $X_{s+1}$  is singular, we miss the opportunity to make a deflation with no work whatsoever. Therefore we decided to take no special action when  $X_{s+1}$  is (nearly) singular.

Our numerical experiments suggest that even when  $X_{s+1}$  is quite ill conditioned, the elimination can usually be carried out without adverse consequences. By this we mean that a new symplectic  $URV$  decomposition is (often) not needed on the next step or any subsequent step.

## 7 Numerical Results

**Example 1:** In [7] the CLM method was used to compute the Hamiltonian real Schur form and the stable invariant subspace for problems 1–19 of the benchmark collection [4]. We ran our code on the same problems with the same choices of parameter values and got comparable results. This is to be expected, as our method reduces to CLM when the block sizes are small. The residual  $r = \|HQ - QH_{\text{final}}\|/\|H\|$  was in the range from  $10^{-13}$  to  $10^{-16}$  on all problems. The only differences between our results and those of CLM were that on problem 11 with parameter  $\epsilon = 0$  and on problems 13 and 14 with  $\epsilon = 10^{-6}$ , our code reported that the matrix has eigenvalues very near the imaginary axis and stopped with a partial Schur decomposition. This is a correct conclusion in all cases, and it was arrived at because of failure to obtain an isotropic

subspace on the last step. If the isotropy tests are disabled, our code produces a Hamiltonian real Schur form in all cases but with somewhat degraded residuals in two of the cases. On problem 11 with  $\epsilon = 0$  we had  $r = 4.6 \times 10^{-9}$ , on problem 13 with  $\epsilon = 10^{-6}$  we had  $r = 4.5 \times 10^{-16}$ , and on problem 14 with  $\epsilon = 10^{-6}$  we had  $r = 7.2 \times 10^{-5}$ . These are about the same as the results reported for CLM in [7]. The relatively poor residuals in the two cases were due to failure to produce a stable invariant subspace that was isotropic to working precision.

**Example 2:** The benchmark collection [4] includes a 20th problem that is quite difficult and is seldom mentioned in the literature. This matrix is of dimension 842 and has ill-conditioned eigenvalues. When we ran our code on this problem, it clustered all of the eigenvalues into a single block. Thus it reduced to the Laub trick in this case. Running MATLAB 7.2, our code failed on this problem because MATLAB’s “schur” command, which is supposed to compute the real Schur form, failed to converge on this matrix. Thus the Laub trick fails for this matrix.

We then balanced the matrix using the symplectic balancing routine from HAPACK [3]. For the balanced matrix the Laub trick was able to compute the stable invariant subspace but had a somewhat large residual  $r = 3.2 \times 10^{-9}$ . This not-very-good result was again a consequence of a failure to produce an invariant subspace that is isotropic to working precision. We should have  $X^T J X = 0$  in theory. What we actually got was  $\|X^T J X\| \approx 10^{-8}$ , which is above the level of roundoff error by a factor of about  $10^8$ . Since the Laub trick’s construction of an orthogonal symplectic transforming matrix relies upon isotropy of the subspace, the lack thereof resulted in a bad residual.

When we ran our block CLM code on the balanced matrix, we were able to extract four quadruples of eigenvalues. The code then decided to lump all of the remaining eigenvalues into a single block. On this (final) step it found that the computed invariant subspace did not satisfy the isotropy condition, so it gave up and returned a partial Hamiltonian Schur form. When we disabled the isotropy test, the code was able to return a complete Hamiltonian real Schur decomposition, but the residual was  $2.1 \times 10^{-10}$ . This relatively poor residual is an expected consequence of the lack of isotropy.

The matrices used in the following examples are available from the third author on request.

**Example 3:** We built a  $40 \times 40$  Hamiltonian matrix with a cluster of 20 ill-conditioned eigenvalues around  $-1$ . Our code diagnosed the cluster and processed it in a single block of size 20. Thus our method reduced to the Laub trick in this case. We obtained a residual  $r = \|HQ - QH_{\text{final}}\|/\|H\| = 5.6 \times 10^{-15}$ . In contrast, the residual produced by CLM was  $r = 8.3 \times 10^{-3}$ .

**Example 4:** We built a  $120 \times 120$  Hamiltonian matrix with the 60 left-half-plane eigenvalues arranged in five tight clusters of 12. Our code processed them as five blocks of size 12 and obtained a residual  $r = 1.3 \times 10^{-14}$ . CLM obtained  $r = 5.9 \times 10^{-1}$  on this problem.

**Example 5:** We built a  $400 \times 400$  matrix with 10 clusters of varying sizes from

6 to 14 in the left half plane, along with many unclustered eigenvalues. Our code partitioned them into 12 blocks, 11 of size 1 and one of size 189. Thus it processed 11 real eigenvalues separately and bunched the remaining 189 into a single block. This was disappointing, as we were hoping that the code would do a better job of blocking the eigenvalues. On the bright side, our code obtained a residual  $r = 1.4 \times 10^{-14}$ . CLM gave  $r = 2.4 \times 10^{-1}$ .

**Example 6:** We built a  $20 \times 20$  Hamiltonian matrix with double eigenvalues at  $\pm i$ . Our code computed a partial Hamiltonian real Schur decomposition and reported that there are four purely imaginary eigenvalues. The residual was  $2.2 \times 10^{-13}$ .

**Example 7:** We generated a  $1000 \times 1000$  Hamiltonian matrix with no eigenvalue clusters and no eigenvalues on the imaginary axis. We ran our code on this matrix three times with different parameter settings. On the first run we set the minimum block size to 1, as we did in all previous examples. Our code built only blocks of size 1 (corresponding to real eigenvalues) and 2 (corresponding to complex conjugate pairs). Thus it reduced to CLM in this case. The residual was  $r = 2.1 \times 10^{-14}$ . In spite of the large size of the matrix, only one symplectic URV decomposition was needed, no supplemental URV decompositions had to be done along the way.

For the second run we set the minimum block size to 40. The actual block sizes that were chosen by the algorithm were 40, 40, 40, 40, 41, 40, 40, 40, 41, 40, 41, and 57, and the final residual was  $r = 2.1 \times 10^{-13}$ .

This run required additional symplectic URV decompositions. In addition to the initial decomposition of the  $1000 \times 1000$  matrix, a second decomposition was required on a submatrix of dimension 760, and additional decompositions were needed at dimensions 518, 358, and 196. This was disappointing and it looks bad, but it is not as bad as it looks. Since the work required for a symplectic URV decomposition is  $O(n^3)$ , we can easily check that the amount of work for the second decomposition is only about 44% of the work for the initial decomposition. For the third through fifth decompositions it is only about 14%, 5%, and 1%, respectively. Thus the total amount of work for all five decompositions was only some 75% more than it was for the first decomposition alone.

Each of the additional symplectic URV decompositions was triggered by a slight failure of invariance of the computed subspace. If we had made the invariant subspace tolerance a bit less stringent, we could have delayed these decompositions and lowered the total number.

The second run was done in mode 1 (see § 6), in which we strictly enforce the minimum block size. The third run was done in mode 2 (with minimum block size 40, as in the second run). In mode 2, if the subspace fails the invariance or isotropy condition, an attempt is made to find a smaller subspace that satisfies these conditions, instead of computing a new symplectic URV decomposition right away. We are willing to violate the specified minimum block size. Running in this mode we were able to avoid any symplectic URV decompositions beyond the first one. The code chose block sizes 40, 40, 40, 38, 40, 38, 28, 35, 27, 3, 5,

and 166. The residual was  $r = 1.9 \times 10^{-12}$ .

Since our code is written in unoptimized MATLAB, we have not listed run times in general. However, for this large example we will report a few times. The computations were done on a computer with a single-core 3.0 GHz Intel Pentium 4 processor with 1.0 MB cache. The run with the minimum block size set to 1 took 128 seconds to complete, excluding the time for the symplectic URV decomposition (27 seconds). This number could undoubtedly be improved a lot by rewriting the code in Fortran or C. When we changed the minimum block size to 40 (running in mode 2), the time was reduced to 25 seconds, again excluding the time for the symplectic URV decomposition. Certainly this significant improvement is at least partly due to improved cache use. For comparison, the Laub trick took 47 seconds total, of which 42 seconds were spent computing an unstructured real Schur decomposition. This is a computation that is done quickly in MATLAB, as it uses efficient LAPACK code.

## 8 Conclusions

We have described and implemented a generalization of the CLM method [7] for computing the Hamiltonian real Schur form of a Hamiltonian matrix. By processing clusters of eigenvalues in blocks, our algorithm is able to solve problems on which the original CLM method fails. Our code checks key subspaces for isotropy and invariance and takes action if these conditions are violated. It either produces the desired Hamiltonian Schur form and/or stable invariant subspace or reports that it was unable to do so. If there are eigenvalues on or very near the imaginary axis, they will be reported. There are many ways in which the code can be tuned. It could turn out that our current settings are far from optimal.

## References

- [1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank. *Matrix Riccati Equations in Control and Systems Theory*. Birkhäuser Verlag, Heidelberg, 2003.
- [2] P. Benner, R. Byers, V. Mehrmann, and H. Xu. A robust numerical method for the  $\gamma$ -iteration in  $h_\infty$  control. *Linear Algebra Appl.*, 425(2-3):548–570, 2007.
- [3] P. Benner and D. Kressner. Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices. *ACM Trans. Math. Software*, 32:352–373, 2006. [www.tu-chemnitz.de/mathematik/hapack/](http://www.tu-chemnitz.de/mathematik/hapack/).
- [4] P. Benner, A. J. Laub, and V. Mehrmann. Benchmarks for the numerical solution of algebraic Riccati equations. *IEEE Contr. Syst. Mag.*, 7:18–28, 1997.



- [5] P. Benner, V. Mehrmann, and H. Xu. A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. *Numer. Math.*, 78:329–358, 1998.
- [6] A. Bunse-Gerstner. Matrix factorizations for symplectic  $QR$ -like methods. *Linear Algebra Appl.*, 83:49–77, 1986.
- [7] D. Chu, X. Liu, and V. Mehrmann. A numerically strongly stable method for computing the Hamiltonian Schur form. *Numer. Math.*, 105:375–412, 2007.
- [8] G. Freiling, V. Mehrmann, and H. Xu. Existence, uniqueness and parametrization of Lagrangian invariant subspaces. *SIAM J. Matrix Anal. Appl.*, 23:1045–1069, 2002.
- [9] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.
- [10] S. Grivet-Talocia. Passivity enforcement via perturbation of hamiltonian matrices. *IEEE Trans. Circuits and Systems*, 51:1755–1769, 2004.
- [11] J. Handy. *The Cache Memory Book*. Academic Press, Second edition, 1998.
- [12] T. Kailath. *Linear Systems*. Information and System Sciences. Prentice-Hall, 1980.
- [13] D. Kressner. *Numerical Methods for General and Structured Eigenproblems*, volume 46 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2005.
- [14] A. J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, 24:913–921, 1979.
- [15] W.-W. Lin and V. Mehrmann und H. Xu. Canonical forms for Hamiltonian and symplectic matrices and pencils. *Linear Algebra Appl.*, 301-303:469–533, 1999.
- [16] V. Mehrmann and H. Xu. Perturbation of purely imaginary eigenvalues of hamiltonian matrices under structured perturbations. Technical Report 410, DFG Research Center, MATHEON, Berlin, Germany, 2007. [url: http://www.matheon.de/](http://www.matheon.de/) , to appear in SIAM J. Matrix Analysis Appl. 2008.
- [17] V. L. Mehrmann. *The Autonomous Linear Quadratic Control Problem*, volume 163 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1991.
- [18] C. C. Paige and C. F. Van Loan. A Schur decomposition for Hamiltonian matrices. *Linear Algebra Appl.*, 41:11–32, 1981.

- [19] A.C.M. Ran and L. Rodman. Stability of invariant Lagrangian subspaces II. In H.Dym, S. Goldberg, M.A. Kaashoek, and P. Lancaster, editors, *Operator Theory: Advances and Applications*, volume 40, pages 391–425. Birkhäuser-Verlag, Basel, Switzerland, 1989.
- [20] C. Schröder and T. Stykel. Passivation of LTI systems. Preprint 368, TU Berlin, MATHEON, Germany, 2007.
- [21] D. S. Watkins. On the reduction of a Hamiltonian matrix to Hamiltonian Schur form. *Electron. Trans. Numer. Anal.*, 23:141–157, 2006.
- [22] D. S. Watkins. *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. SIAM, Philadelphia, 2007.
- [23] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, NJ, 1995.